ARGONNE ATPESC2025 EXTREME-SCALE COMPUTING

Deep Learning for Accelerating Uncertainty Visualization and Visualizing Model Uncertainty

Mengjiao Han
Postdoc Appointee
Leadership Computing Facility, Argonne National Laboratory

Understanding Uncertainty in Scientific Computing

What is Uncertainty?

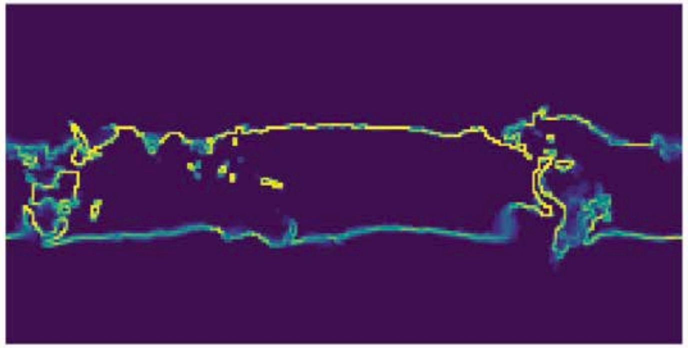
- **Uncertainty** represents the degree of confidence, or lack thereof, in data, models, or predictions.
- It arises from:
  - Incomplete data
  - Approximate models
  - Measurement noise
  - Simulation variability

Types of Uncertainty

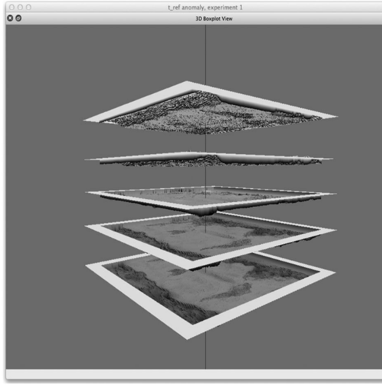
- **Aleatoric (Data-driven)**: Inherent randomness in the system (e.g., noise).
- **Epistemic (Model-driven)**: Uncertainty due to limited knowledge or approximation errors (e.g., neural network predictions)

Agenda

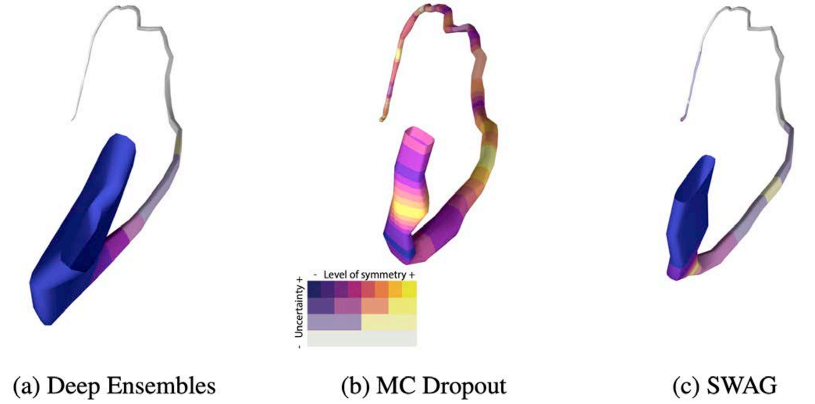
- Accelerating Uncertainty Visualization with Deep Learning
  - Level-set positional uncertainty
  - Surface boxplots
- Visualizing Uncertainty in Deep Learning-Based Particle Tracing
  - Uncertainty-Aware Neural Pathline Tracing
  - Uncertainty tube visualization



Visualization of positional uncertainty of level sets for a wind temperature dataset [Han et al. 2022]



Surface Boxplots [Genton et al. 2014]



(a) Deep Ensembles (b) MC Dropout (c) SWAG

Visualization of particle tracing neural network uncertainty [under review]

ARGONNE
ATPESC2025
EXTREME-SCALE COMPUTING

Part 1: Deep Learning to Accelerate Uncertainty Visualization

Positional Uncertainty of Level Sets for Ensemble Simulations

Collaborated with Tushar Athawale[1], David Pugmire[1], Chris R Johnson[2]

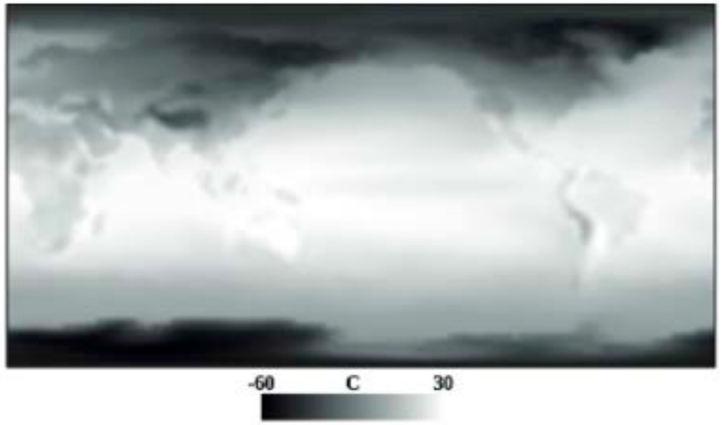
Surface Boxplots for Ensemble Simulations

Collaborated with Tushar Athawale[1], Jixian Li[2], Chris R Johnson[2]

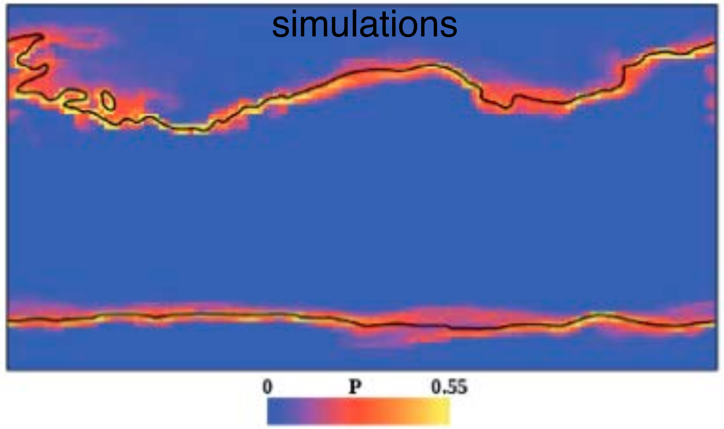
1. Oak Ridge National Laboratory 2. Scientific Computing and Imaging Institute

Uncertainty of Ensembles: Positional Uncertainty of Level Sets

- **Level sets (isosurfaces)** represent important features in scalar fields
- In ensembles, level set positions vary across members
- Positional uncertainty: how likely is the isosurface to pass through a voxel?
- Traditional method: **Probabilistic Marching Cubes (PMC)**
  - Uses Monte Carlo sampling
  - Very expensive!

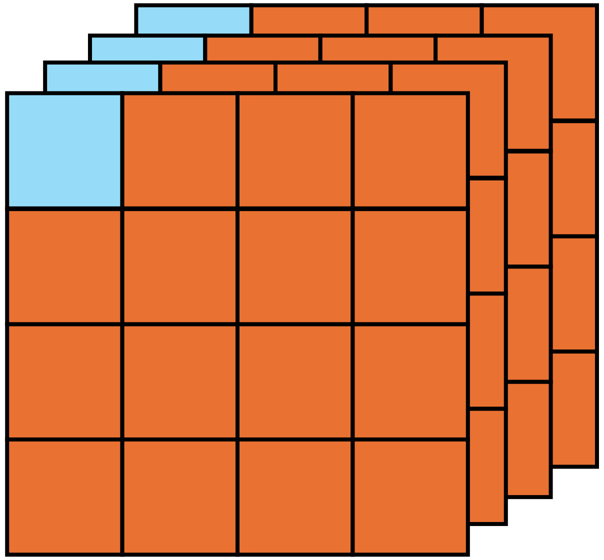


The ensemble means of the temperature field from climate simulations



The level-set crossing probabilities for temperature = 0

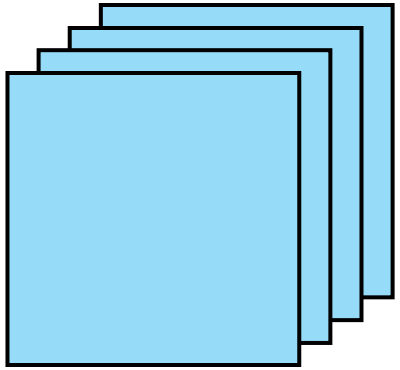
Probabilistic Marching Cubes (PMC): Use Monte Carlo Sampling



M ensemble members

$Y_0 = [y_0^0, y_0^1,\ y_0^2, \ldots, y_0^M]$

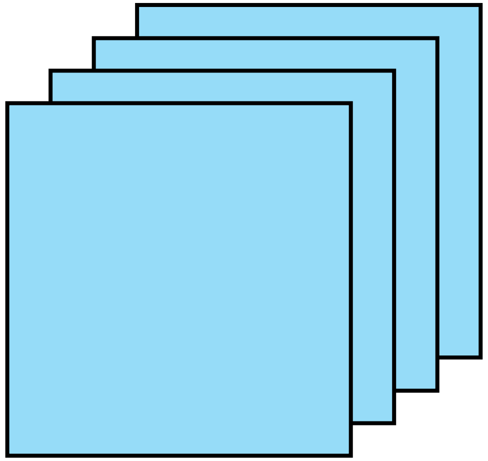
$Y_1 = [y_1^0, y_1^1,\ y_1^2, \ldots, y_1^M]$



$Y_2 = [y_2^0, y_2^1,\ y_2^2, \ldots, y_2^M]$

$Y_3 = [y_3^0, y_3^1,\ y_3^2, \ldots, y_3^M]$

Probabilistic Marching Cubes (PMC): Use Monte Carlo Sampling



$Y_i = [y_i^0, y_i^1, \; y_i^2, \ldots, y_i^M]$
$i = 0, 1, 2, 3$

M ensemble members

Means:
$\mu = [\mu_0, \mu_1, \mu_2, \mu_3]$

Covariance Matrix:

$$Cov_{i,j} = \frac{1}{M-1} \sum_{1}^{M} (y_i^m - \mu_i)(y_j^m - \mu_j)$$

Where $i, j = 0, 1, 2, 3$

Drawing **r samples** from a multivariate Gaussian distribution

LCP $p = \frac{k}{r}$ if a level set passes through k samples

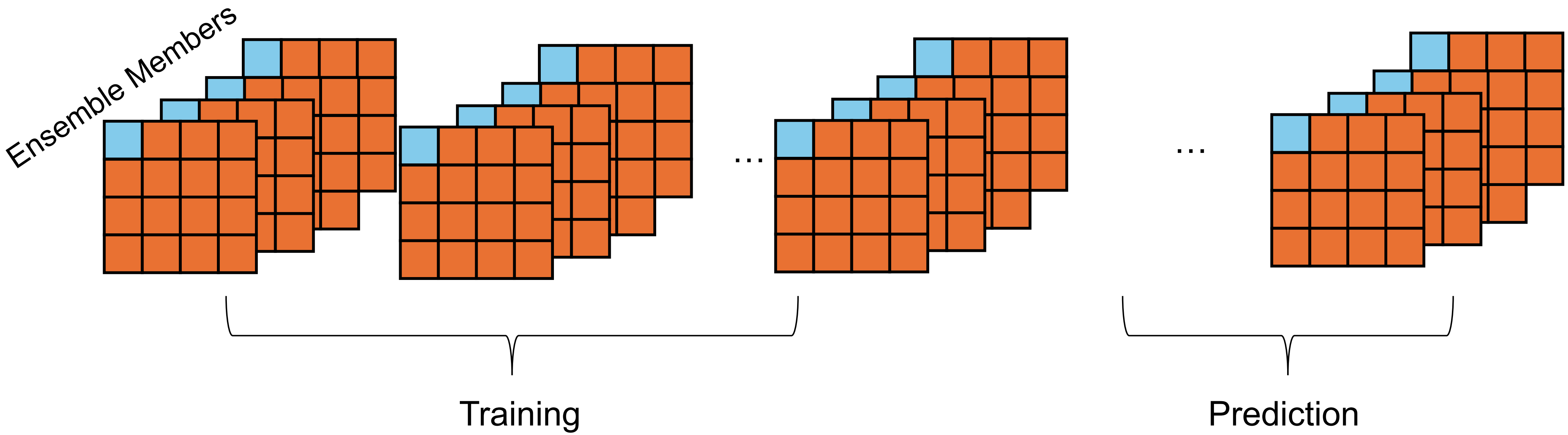
Accelerating PMC with Deep Learning

- Goal: replace Monte Carlo sampling with a fast neural prediction
- Predict **Level-Crossing Probability (LCP)** for ensemble datasets
- **Key idea:**
  - Inputs: mean, variance, isovalue
  - Output: LCP (probability that the level set crosses a voxel)
- One training sample represents one grid cell with a one-dimensional vector of size 16

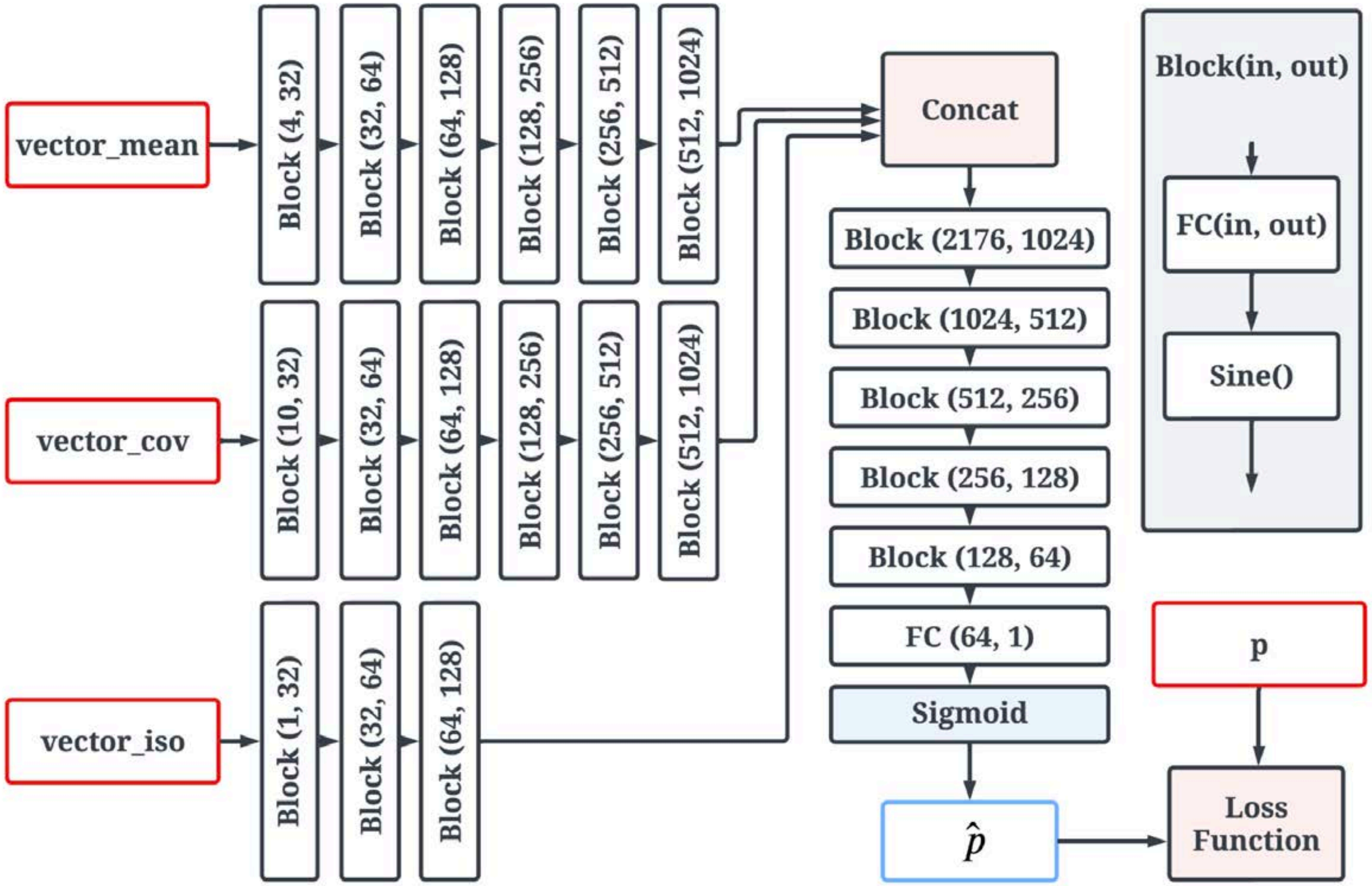
$$[\underbrace{\mu_0, \mu_1, \mu_2,}_{\substack{\text{Means}\\ \text{(vector\_mean)}}} \mu_3, \underbrace{\sigma_0^2, \sigma_1^2, \sigma_2^2, \sigma_3^2, Cov_{0,1}, Cov_{0,2}, Cov_{0,3}, Cov_{1,2}, Cov_{1,3}, Cov_{2,3}}_{\substack{\text{Variances and Covariance}\\ \text{(vector\_cov)}}},\ \underbrace{s}_{\substack{\text{isovalue}\\ \text{(vector\_iso)}}},\ \underbrace{\color{red}{p}}_{\textbf{LCP}}]$$

Training Data Generation

- Training samples are generated from PMC on time-varying ensemble datasets

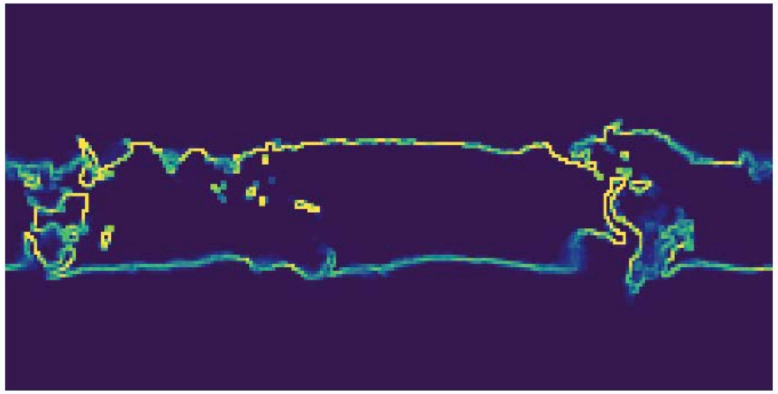


Network Architecture

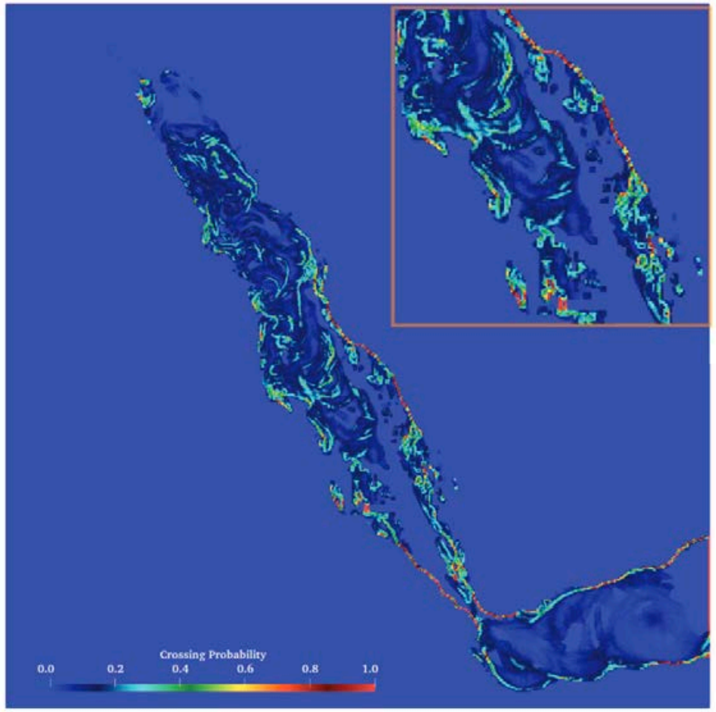


Datasets

- Wind [240 x 121]:
  - 45 time steps, 15 members for each time step
  - 17 time steps for training
- Red Sea [500 x 500]:
  - 60 time steps, 50 members for each time step
  - 10 time steps for training

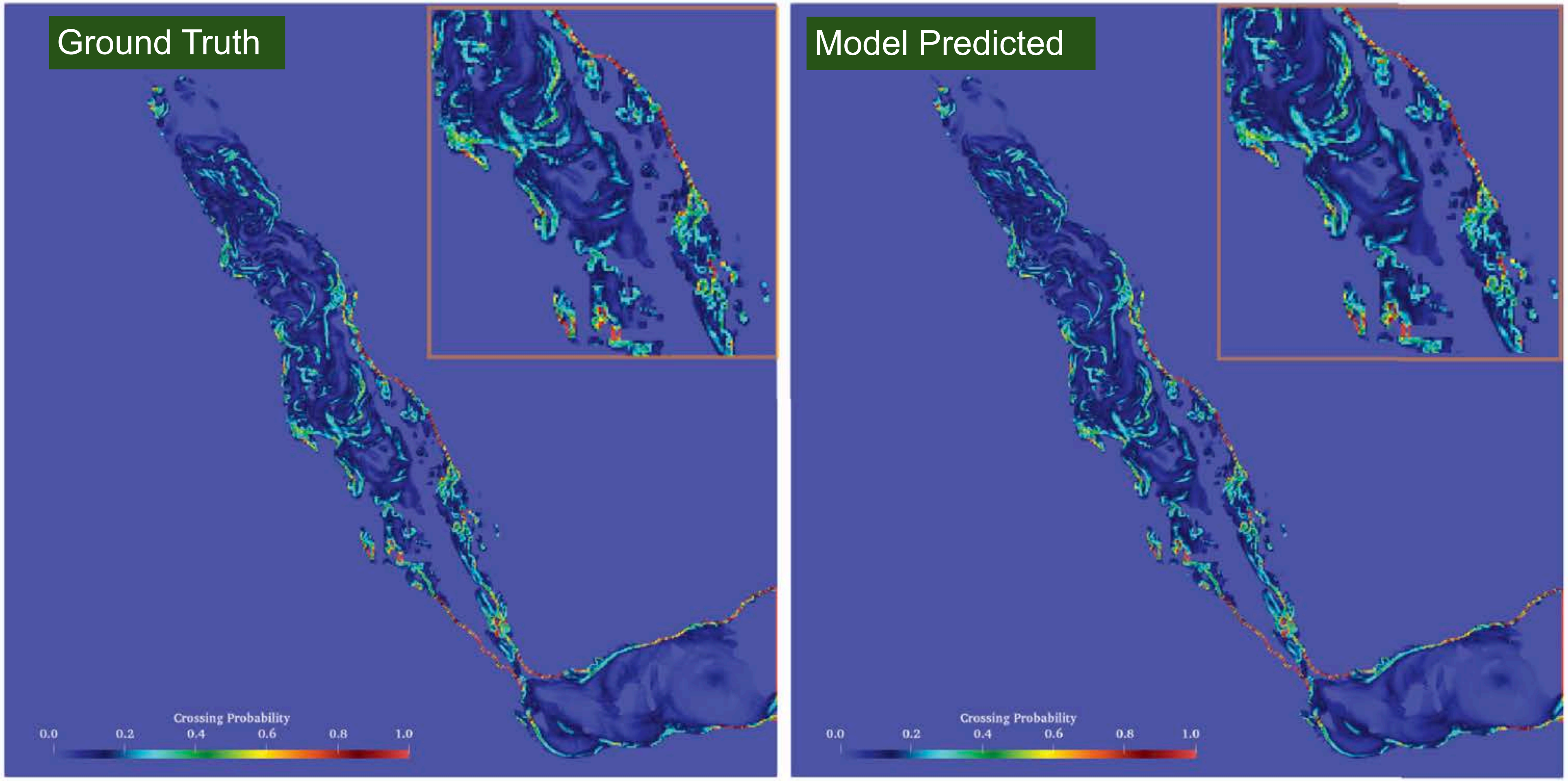


(a) Visualization of level-set corssing probability for temperature in Wind dataset at time step 22 with iso-value 0.8



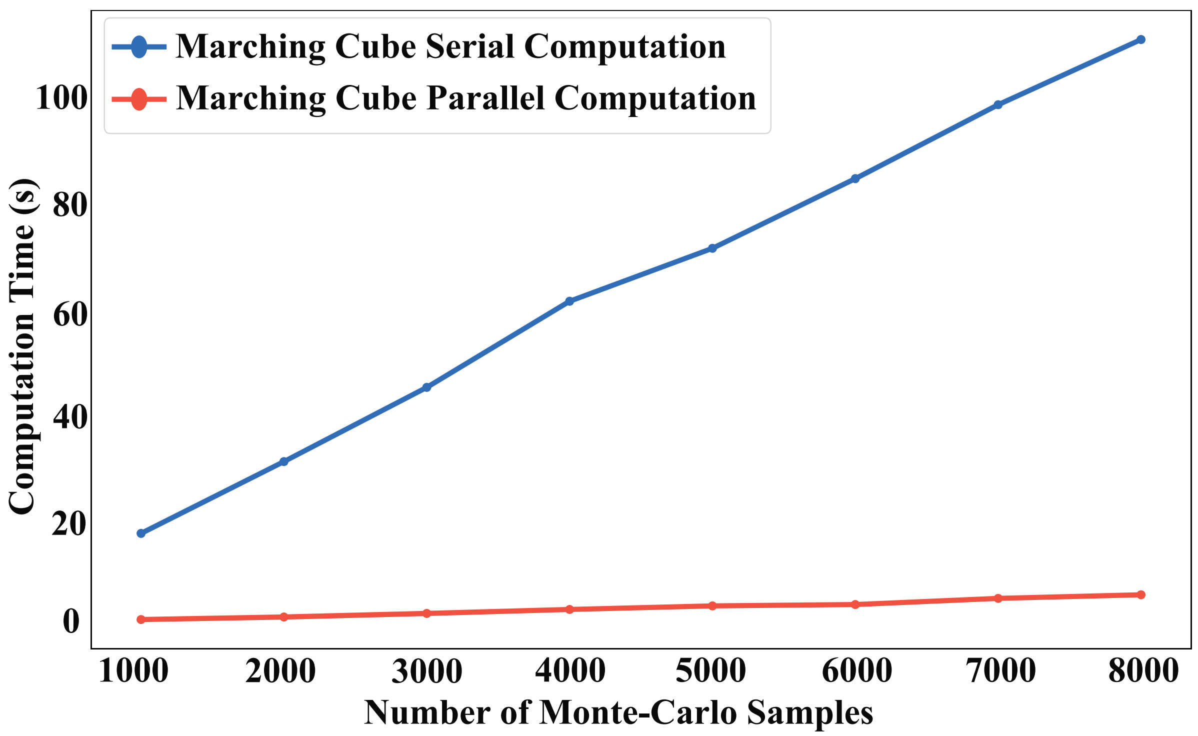
(b) Visualization of level-set corssing probability for velocity in Red Sea dataset at time step 51 with iso-value 0.1

Results: Predicted LCPs are Indistinguishable from the Ground Truth

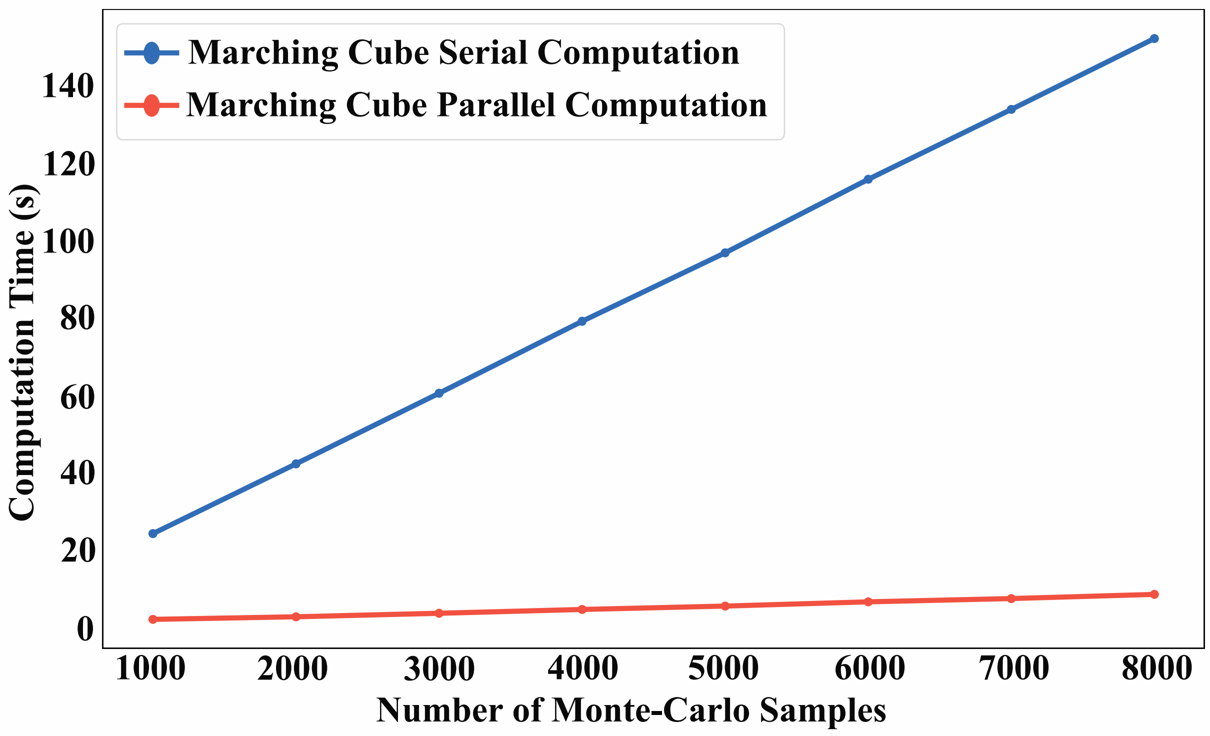


Visualizations of the level-set crossing probability for isovalue 0.1 in the Red Sea dataset.

Results: Parallel Computation is 17X Faster than Serial Computation

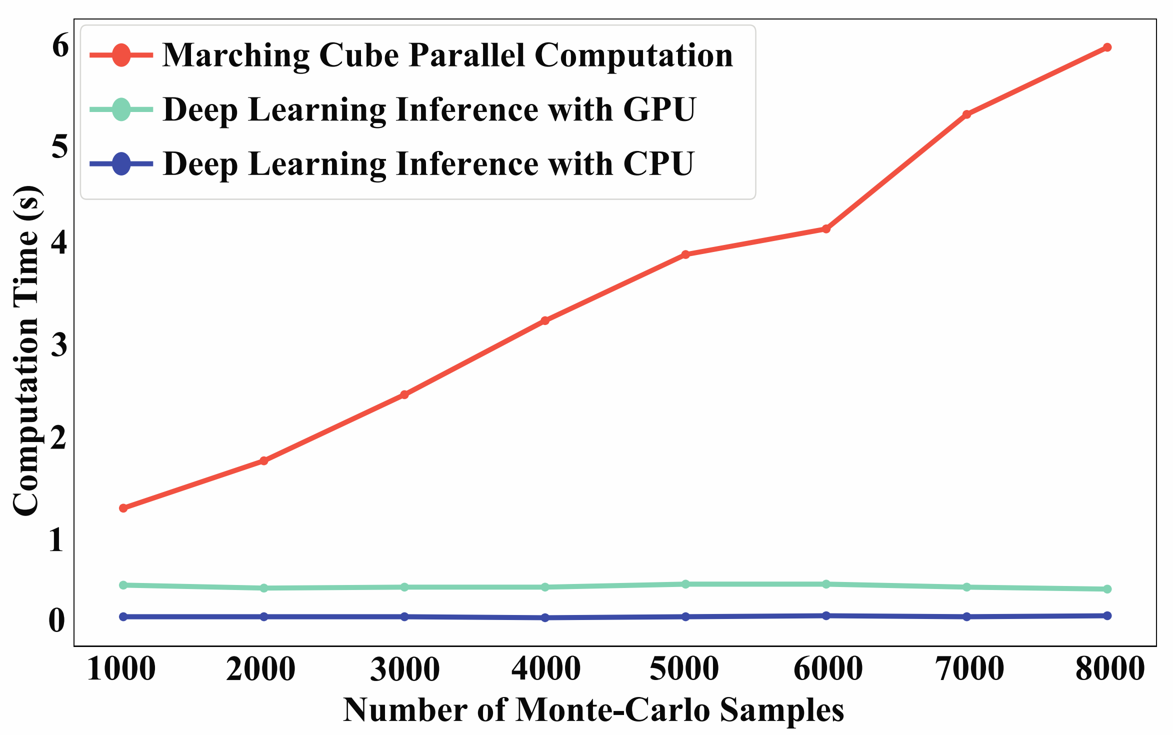


Wind. Time step = 33, isovalue = 0.2

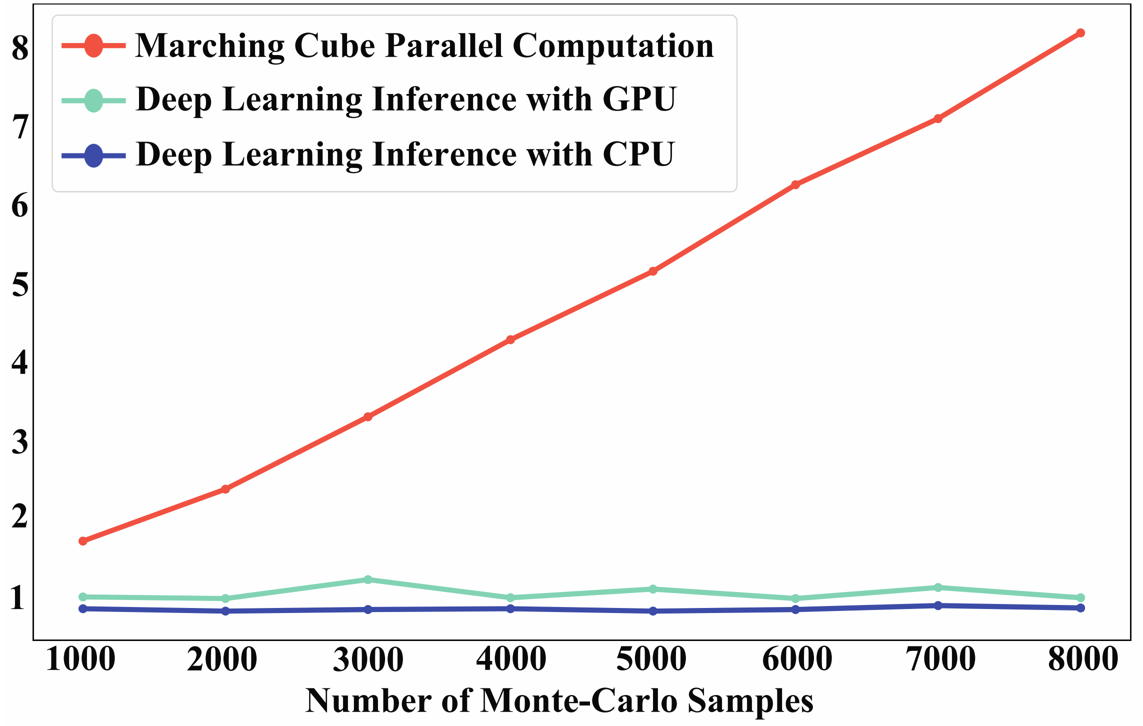


Red Sea. Time step = 53, isovalue = 0.1

Results: 10X Faster than the Parallel PMC

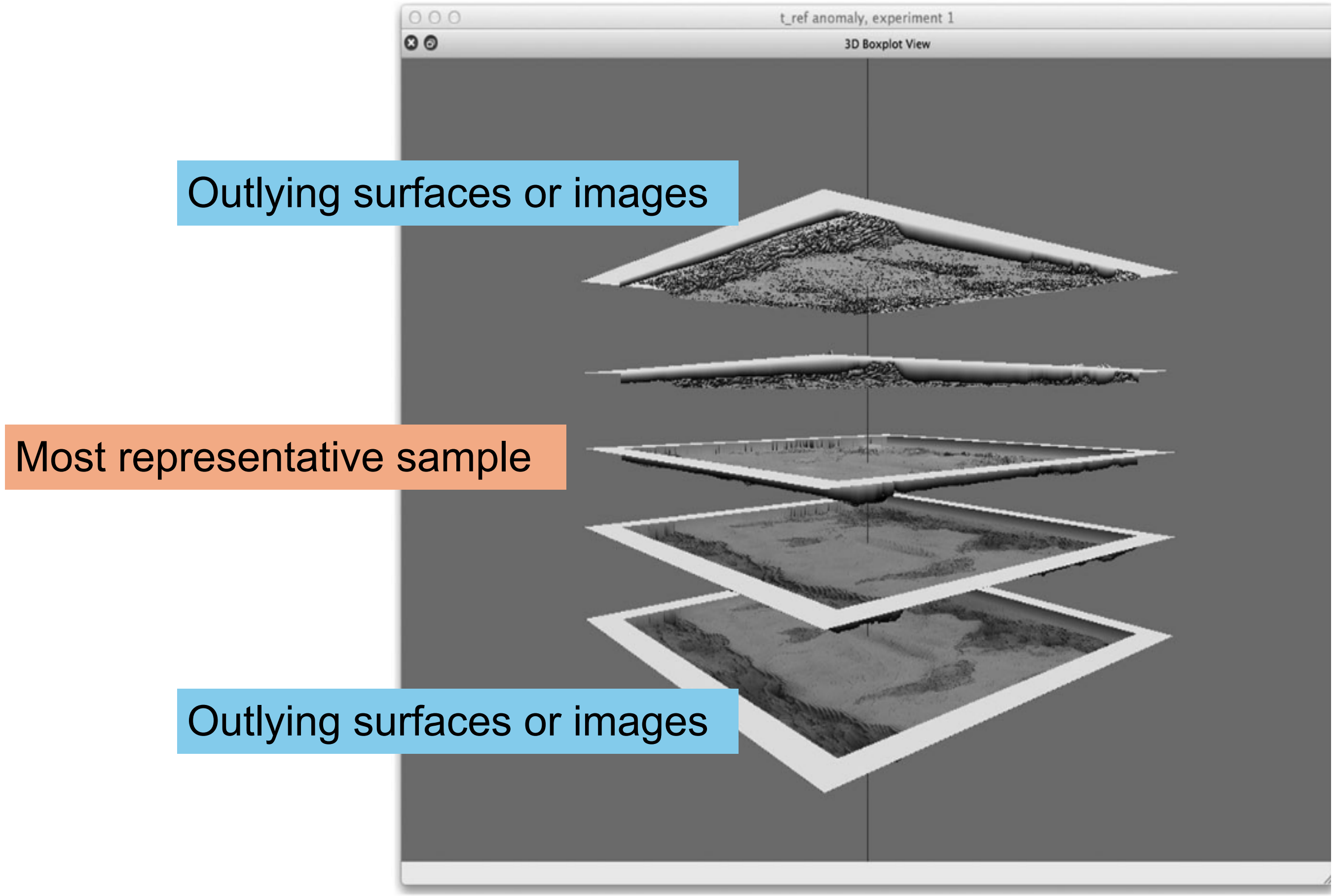


Wind. Time step = 33, isovalue = 0.2



Red Sea. Time step = 53, isovalue = 0.1

Extending the Idea: Surface Boxplots for Ensemble Fields



- Surface boxplots compute central representation sample and outliers in ensembles
- Traditional approach: depth-based ranking (e.g., band depth)

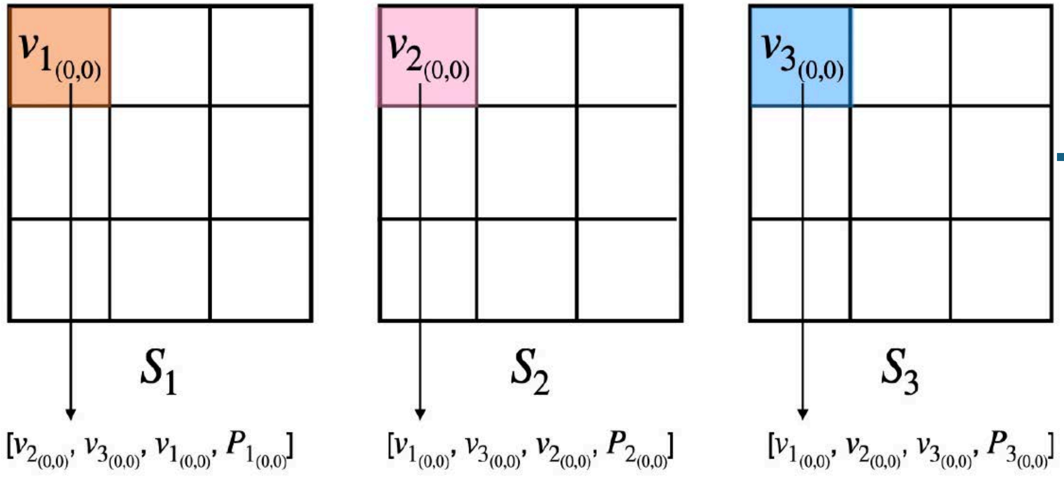
Cost of Computation Increases Cubically with the Number of Ensemble Members

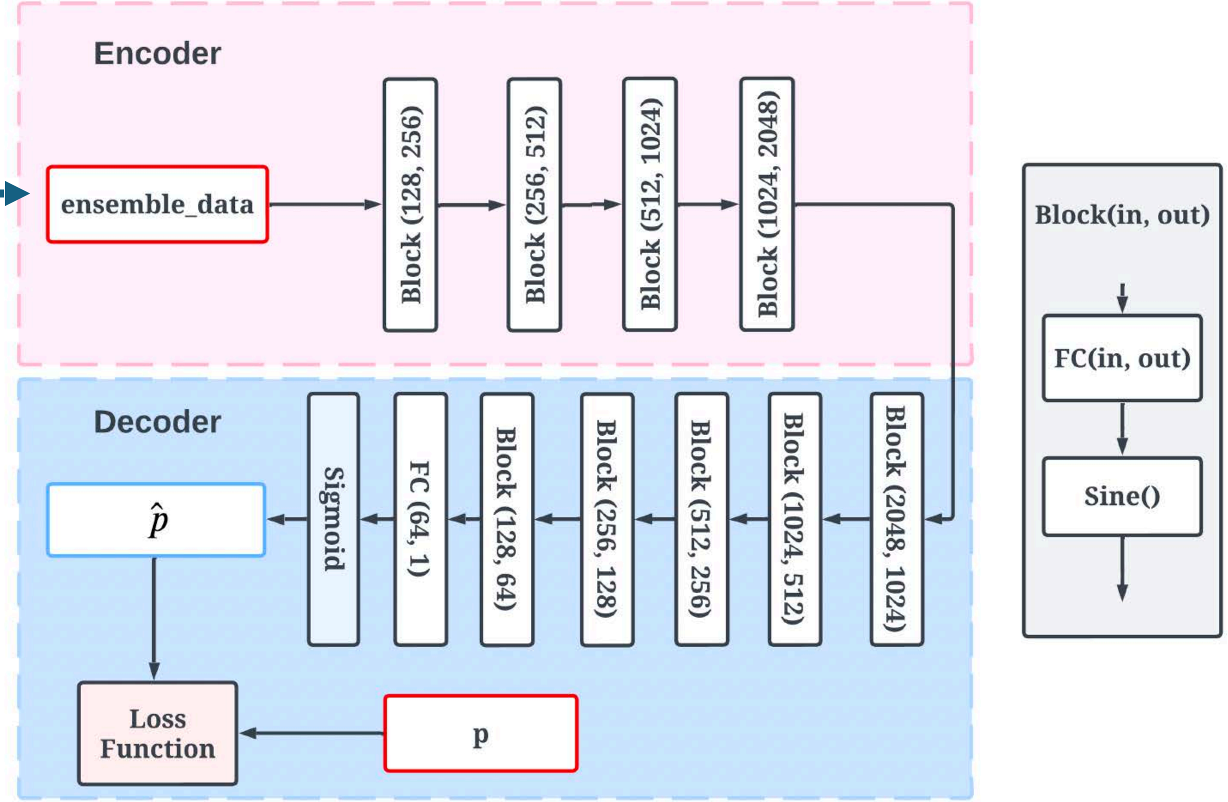
- Cost increases cubically with ensemble size
- Can we again replace this with a neural predictor?

```
for each member in N ensembles:
    volume = ensembles[index]
    combinations = C(n-1, 2)
    for combination in combinations:
      volume0 = combination[0]
      volume1 = combination[1]
      for each voxel in volume:
        # Check whether the current data value
        # falls within the range defined by volume0 and volume1
        if in the band: depth += 1
```

The main three-nested for loop in the algorithm

MLP-Based Neural Network





The figure shows three ensemble members at a single time step. Each sample includes the data values from the ensemble members along with the depth $P_{i_{(x,y)}}$

Prediction Accuracy: Rank Preserved with Occasional Order Flips

#Time Step		Depth Order	Level of Rank Preservation
#3	GT	[35, 9, 5, 39, 24, 29, 20, 34, 44, 22, 14, 4, 31, 42, 18, 49, 1, 33, 7, 45, 46, 16, 28, 19, 6, 48, 13, 43, 21, 37, 2, 47, 25, 0, 8, 40, 15, 30, 10, 11, 36, 32, 26, 27, 17, 12, 3, 38, 41, 23]	99.84%
	Pred	[35, 9, 5, 39, 24, 29, 20, 34, 44, 22, 14, 4, 31, 42, 18, 49, 1, 33, 7, 45, 46, 16, 28, 19, 48, 6, 13, 43, 21, 37, 2, 47, 25, 0, 8, 40, 15, 30, 10, 11, 36, 32, 26, 17, 27, 12, 3, 38, 41, 23]	
#9	GT	[9, 5, 39, 29, 35, 20, 24, 44, 42, 4, 14, 49, 34, 18, 31, 22, 33, 16, 28, 46, 45, 48, 21, 43, 19, 47, 13, 1, 7, 6, 30, 0, 8, 37, 25, 11, 40, 27, 3, 15, 26, 12, 10, 2, 38, 41, 36, 32, 17, 23]	99.92%
	Pred	[9, 5, 39, 29, 35, 20, 24, 44, 42, 4, 14, 49, 34, 18, 31, 22, 33, 16, 28, 46, 45, 48, 21, 43, 47, 19, 13, 1, 7, 6, 30, 0, 8, 37, 25, 11, 40, 27, 3, 15, 26, 12, 10, 2, 38, 41, 36, 32, 17, 23]	
#17	GT	[35, 29, 5, 9, 39, 20, 14, 4, 44, 42, 24, 49, 22, 18, 47, 28, 43, 16, 13, 48, 46, 21, 45, 33, 31, 34, 8, 30, 7, 11, 25, 27, 19, 12, 1, 6, 41, 26, 38, 0, 37, 40, 3, 10, 15, 2, 32, 23, 36, 17]	99.92%
	Pred	[35, 29, 5, 9, 39, 20, 14, 4, 44, 42, 24, 49, 22, 18, 47, 28, 43, 16, 13, 48, 46, 21, 45, 33, 31, 34, 8, 30, 7, 11, 25, 27, 19, 12, 6, 1, 41, 26, 38, 0, 37, 40, 3, 10, 15, 2, 32, 23, 36, 17]	
#19	GT	[35, 29, 5, 9, 20, 39, 4, 42, 14, 44, 24, 49, 22, 18, 28, 47, 43, 48, 13, 16, 21, 45, 31, 46, 34, 8, 30, 11, 33, 7, 27, 12, 25, 1, 10, 6, 37, 38, 0, 41, 19, 26, 40, 3, 15, 2, 32, 23, 36, 17]	99.76%
	Pred	[35, 29, 5, 9, 20, 39, 4, 42, 44, 14, 24, 49, 22, 18, 47, 28, 43, 48, 13, 16, 21, 45, 31, 46, 34, 8, 30, 11, 33, 7, 27, 12, 25, 10, 1, 6, 37, 38, 0, 41, 19, 26, 40, 3, 15, 2, 32, 23, 36, 17]	
#22	GT	[35, 29, 9, 20, 39, 5, 4, 42, 44, 14, 24, 18, 49, 47, 28, 43, 22, 16, 21, 48, 13, 46, 45, 34, 8, 7, 30, 11, 31, 33, 27, 12, 25, 37, 38, 1, 41, 0, 10, 3, 19, 40, 6, 26, 15, 2, 32, 23, 36, 17]	99.76%
	Pred	[35, 9, 29, 20, 39, 5, 4, 42, 44, 14, 24, 18, 49, 47, 28, 43, 22, 16, 21, 48, 13, 46, 45, 34, 8, 7, 30, 11, 31, 33, 27, 12, 25, 37, 38, 1, 41, 10, 0, 3, 40, 19, 6, 26, 15, 2, 32, 23, 36, 17]	
#30	GT	[20, 35, 29, 9, 5, 4, 39, 42, 44, 14, 49, 47, 18, 28, 24, 43, 21, 13, 22, 16, 48, 46, 8, 45, 11, 34, 27, 31, 30, 12, 33, 7, 38, 25, 41, 37, 1, 0, 3, 10, 6, 26, 40, 19, 15, 23, 32, 36, 2, 17]	99.67%
	Pred	[20, 35, 29, 9, 5, 4, 39, 42, 44, 14, 49, 18, 47, 28, 24, 43, 21, 13, 22, 16, 48, 46, 8, 45, 11, 34, 27, 31, 30, 12, 33, 7, 38, 25, 37, 41, 0, 1, 3, 10, 6, 40, 26, 19, 15, 23, 32, 36, 2, 17]	
#36	GT	[35, 29, 20, 5, 4, 9, 39, 42, 44, 47, 14, 21, 49, 43, 24, 28, 18, 13, 22, 16, 48, 8, 11, 46, 27, 45, 30, 34, 12, 33, 7, 31, 41, 38, 37, 0, 25, 1, 10, 26, 3, 6, 19, 40, 15, 23, 36, 32, 2, 17]	99.92%
	Pred	[35, 29, 20, 5, 4, 9, 39, 42, 44, 47, 14, 21, 49, 43, 24, 28, 18, 22, 13, 16, 48, 8, 11, 46, 27, 45, 30, 34, 12, 33, 7, 31, 41, 38, 37, 0, 25, 1, 10, 26, 3, 6, 19, 40, 15, 23, 36, 32, 2, 17]	
#42	GT	[35, 4, 29, 5, 20, 42, 9, 39, 44, 47, 21, 49, 14, 28, 43, 24, 13, 22, 18, 16, 8, 48, 11, 46, 30, 27, 45, 12, 38, 37, 7, 33, 41, 34, 31, 1, 25, 40, 26, 10, 0, 6, 3, 23, 15, 19, 36, 32, 2, 17]	99.84%
	Pred	[35, 4, 29, 5, 20, 42, 9, 39, 44, 47, 21, 49, 14, 28, 43, 24, 13, 22, 18, 16, 48, 8, 11, 46, 30, 27, 45, 12, 38, 37, 7, 33, 34, 41, 31, 1, 25, 40, 26, 10, 0, 6, 3, 23, 15, 19, 36, 32, 2, 17]	
#47	GT	[29, 4, 35, 5, 20, 42, 9, 44, 39, 21, 47, 14, 49, 43, 28, 13, 22, 24, 18, 16, 48, 8, 11, 46, 30, 12, 27, 45, 38, 7, 31, 37, 41, 33, 25, 34, 0, 40, 26, 1, 3, 10, 6, 23, 15, 19, 32, 36, 2, 17]	99.76%
	Pred	[29, 4, 35, 20, 5, 42, 9, 44, 39, 21, 47, 14, 49, 43, 28, 13, 22, 24, 18, 16, 48, 8, 11, 46, 30, 12, 27, 45, 38, 7, 37, 31, 41, 33, 25, 34, 40, 0, 26, 1, 3, 10, 6, 23, 15, 19, 32, 36, 2, 17]	
#56	GT	[29, 4, 42, 20, 5, 44, 35, 21, 9, 47, 39, 14, 49, 28, 43, 22, 18, 16, 13, 24, 8, 11, 46, 48, 30, 27, 45, 7, 41, 38, 37, 12, 33, 0, 31, 25, 40, 3, 26, 1, 23, 34, 10, 15, 6, 19, 2, 36, 32, 17]	99.67%
	Pred	[29, 4, 42, 20, 5, 44, 35, 21, 9, 47, 39, 14, 49, 28, 43, 22, 18, 16, 13, 8, 24, 11, 46, 48, 30, 27, 45, 7, 41, 38, 37, 12, 33, 0, 31, 25, 40, 3, 23, 26, 1, 34, 10, 15, 6, 19, 2, 36, 32, 17]	

Comparison between the order using the depth predicted from NN (pred) and the ground truth (GT) for Red Sea dataset

Computational Performance: 15X Speed-Up with GPU and 6X with CPU

Datasets	#Ensembles	Traditional Approach (CPU)	DL (CPU)	DL (GPU)
Wind	15	0.336 s	11.50 s	5.85 s
Red Sea	50	245.81 s	40.66 s	16.61 s

Computational time of using the traditional approach with CPU, our trained deep learning model with both CPU and GPU

Conclusion and Future Work

- First study to apply a deep neural network for uncertainty computation
  - Fast inference
  - Visual quality preservation
- Future:
  - Generalized models across datasets!
  - Integration with HPC platforms like Aurora!

Picture of Aurora exascale supercomputer at Argonne

Reference: [1] Han, Mengjiao, et al. "Accelerated probabilistic marching cubes by deep learning for time-varying scalar ensembles." 2022 IEEE Visualization and Visual Analytics (VIS). IEEE, 2022.
[2] Han, Mengjiao, et al. "Accelerated Depth Computation for Surface Boxplots with Deep Learning." 2024 IEEE Workshop on Uncertainty Visualization: Applications, Techniques, Software, and Decision Frameworks. IEEE, 2024.
Github: [1] https://github.com/MengjiaoH/DeepLearning_LCP
[2] https://github.com/MengjiaoH/SurfaceBoxplot_CXX

ARGONNE
ATPESC2025
EXTREME-SCALE COMPUTING

Part 2: Quantifying and Visualizing Uncertainty in Particle Tracing Neural Network

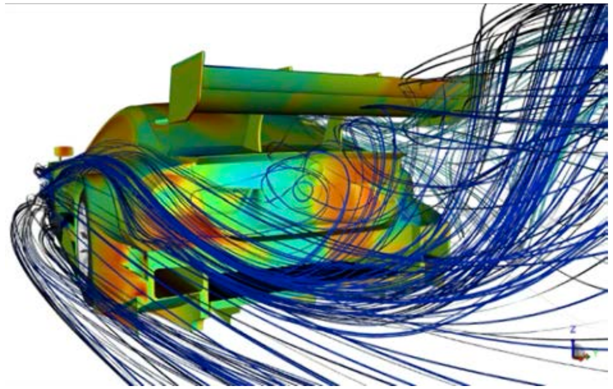
Visualizing the uncertainty of a neural-network-based particle tracing model

Collaborated with Jixian Li[1] and Timbwaoga Aime Judicael Ouermi[1]

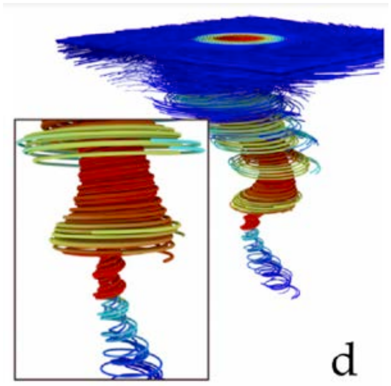
1. Scientific Computing and Imaging Institute

The Need for Uncertainty-Aware Models in Particle Tracing Neural Network

- Visualizing particle trajectories are critical in understanding fluid dynamics
  - I/O limitations
  - Memory constraints
- Neural networks (NNs) offer computational efficiency in predicting trajectories.
- Visualizing prediction uncertainty is crucial for reliable decision-making



BMW Motorsport, Computational Fluid Dynamics simulation, BMW M4 DTM. 2017



Visualization of streamlines of a tornado [Han et al. 2019]

References

- Deep Particle Tracker: Automatic Tracking of Particles in Fluorescence Microscopy Images Using Deep Learning [Spilger et al 2018]
- Exploratory Lagrangian-based Particle Tracing using Deep Learning [Han et al. 2022]
- Neural Flow Map Reconstruction [Sahoo et al. 2022]
- Interactive Visualization of Time-Varying Flow Fields
- Using Particle Tracing Neural Networks [Han et al. 2024]

Challenges in Visualizing Uncertainty in Flow Field
Dynamic Uncertainty is Difficult to Visualize

- cluttered

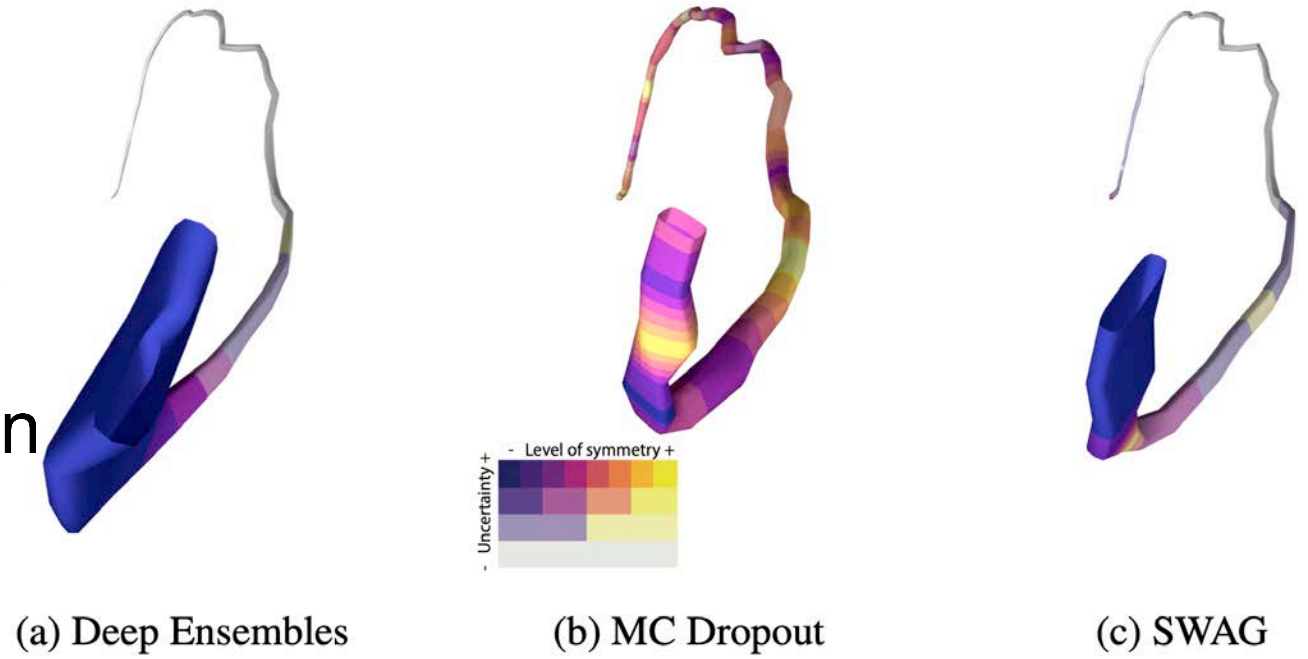
(a) Spaghetti plot

(b) Circular tube

- Assume symmetric uncertainty

Research Objectives

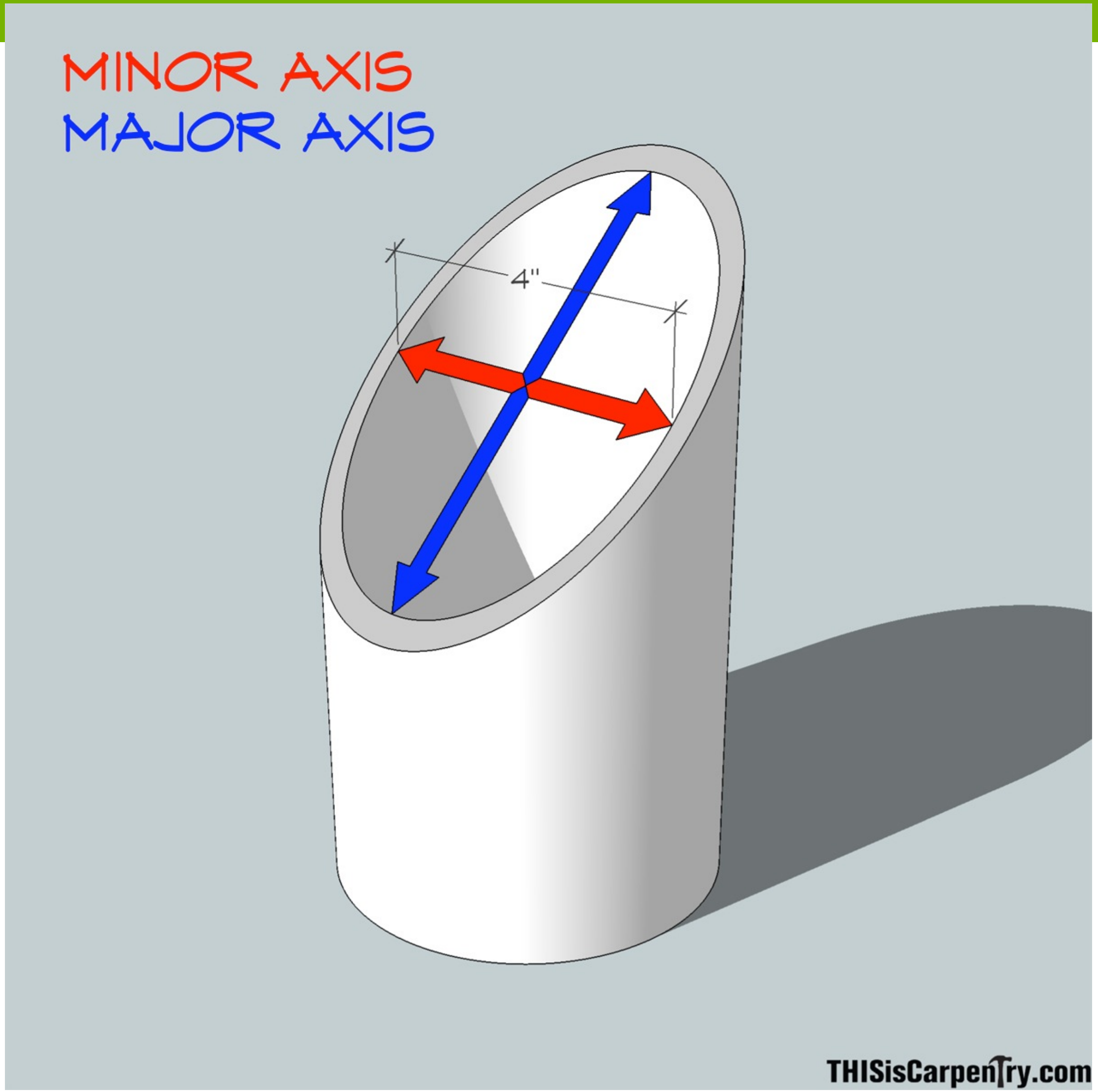
- Develop **uncertainty tube**: an intuitive, efficient uncertainty visualization method for flow data
- Integrate uncertainty quantification methods for neural networks:
  - Deep Ensembles
  - Monte Carlo (MC) Dropout
  - Stochastic Weight Averaging-Gaussian (SWAG)



Uncertainty tube visualizations of one pathline using (a) deep ensemble, (b) MC dropout, and (c) Stochastic Weight Averaging-Gaussian (SWAG) methods.

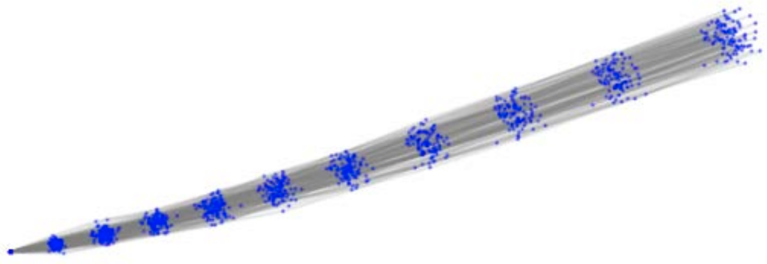
Uncertainty Tube: Visualizing Directional Uncertainty

- Addresses **non-symmetric uncertainty** effectively.
- Employs **superellipse** geometry to represent uncertainty.

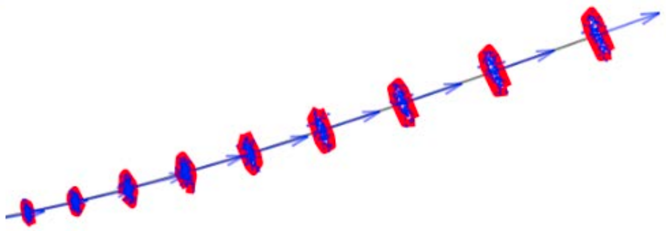


Uncertainty Tube: Visualizing Directional Uncertainty

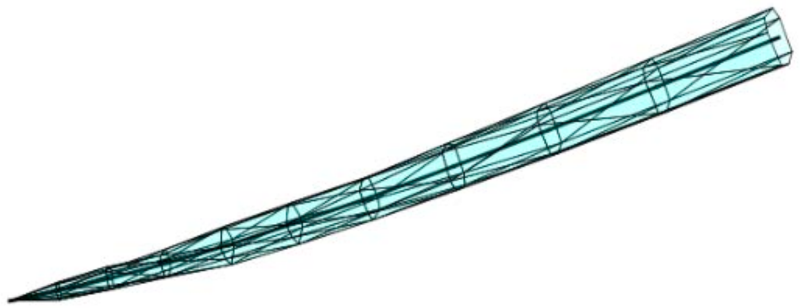
- Start with an ensemble of N pathlines from t to $t + \delta$
- Project each point onto the orthogonal plane of mean point
- Compute covariance and eigen-decomposition and generate superellipse to visualize uncertainty distribution
- Align tubes to minimize twisting and distortion



(A) Raw Sample Trajectories



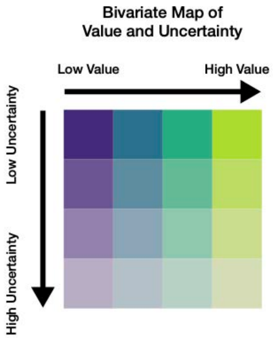
(B) Uncertainty Superellipse



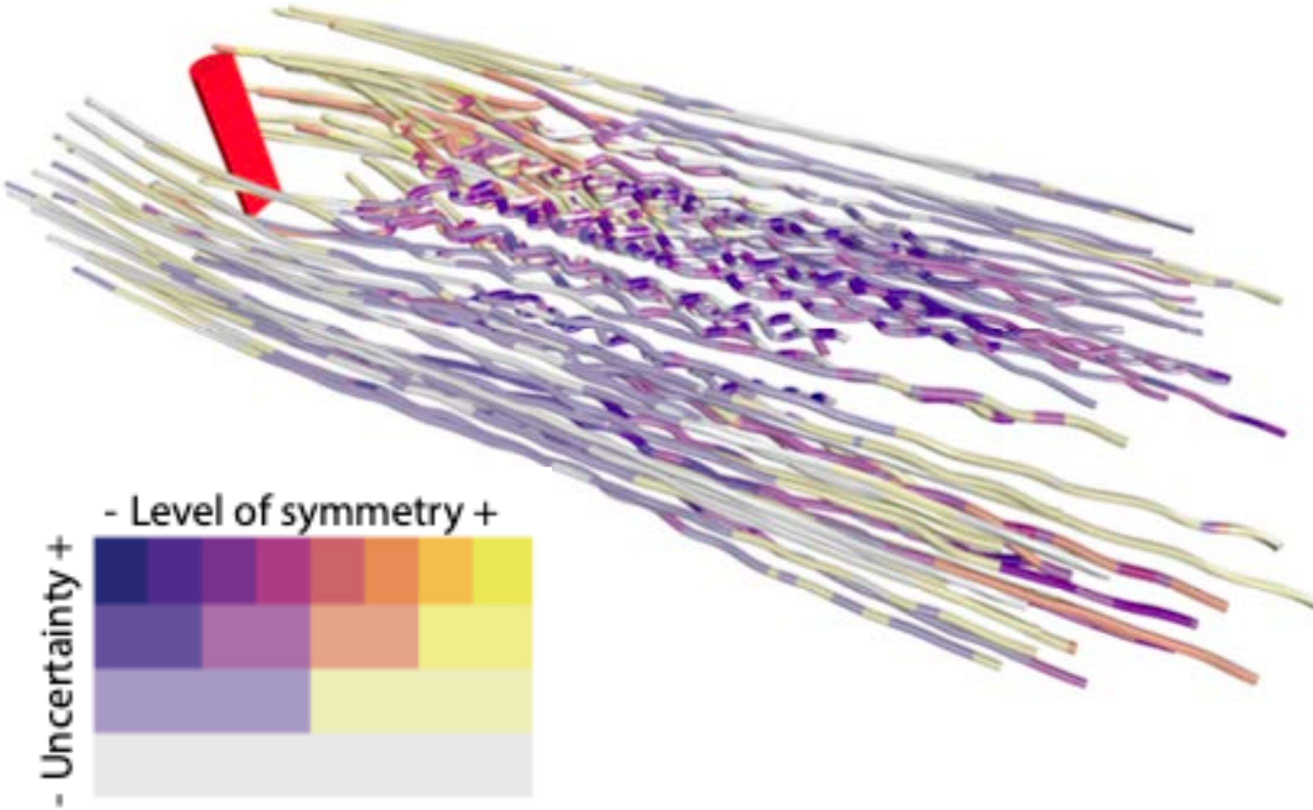
(C) UncertaintyTube Mesh after Alignment

Uncertainty Tube: Color Representation

- Inspired by value-suppressing uncertainty palettes (VSUP)



- Color map:
  - Gray: Low uncertainty. The colormap does not distinguish between the levels of symmetry.
  - Blue: High asymmetric uncertainty.
  - Yellow: High symmetric uncertainty.



Asymmetric uncertainty colormap in our method

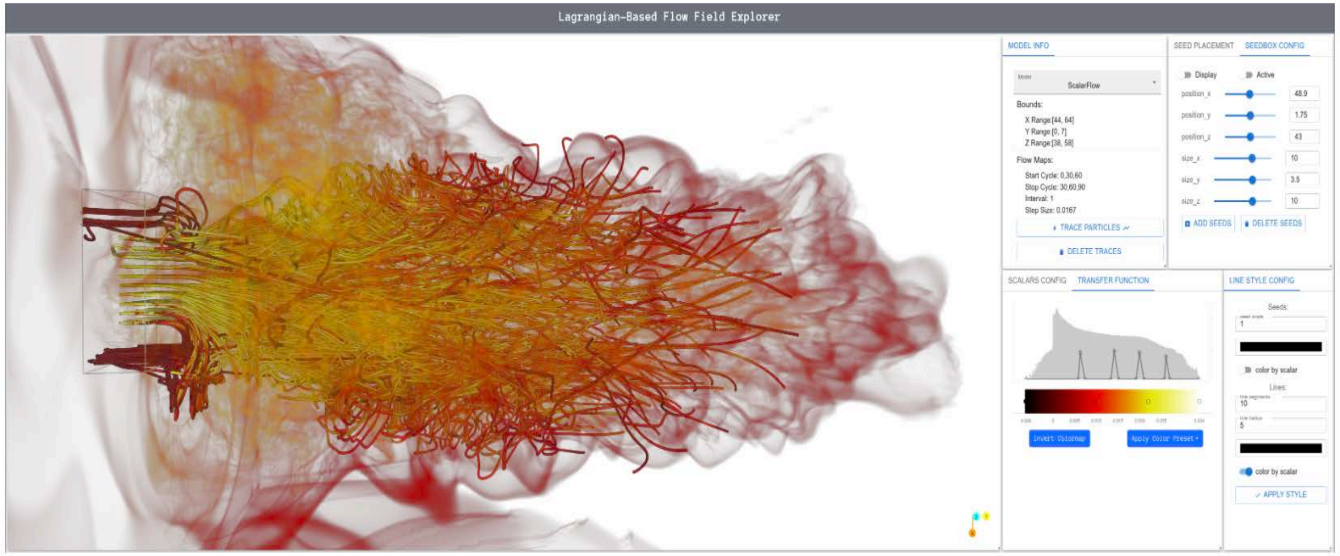
Uncertainty Tube: Computational Efficiency

The performance (seconds) on AMD Ryzen Threadripper 3970X 32-Core Processor using 32 cores. 50 uncertainty samples per trajectory are used.

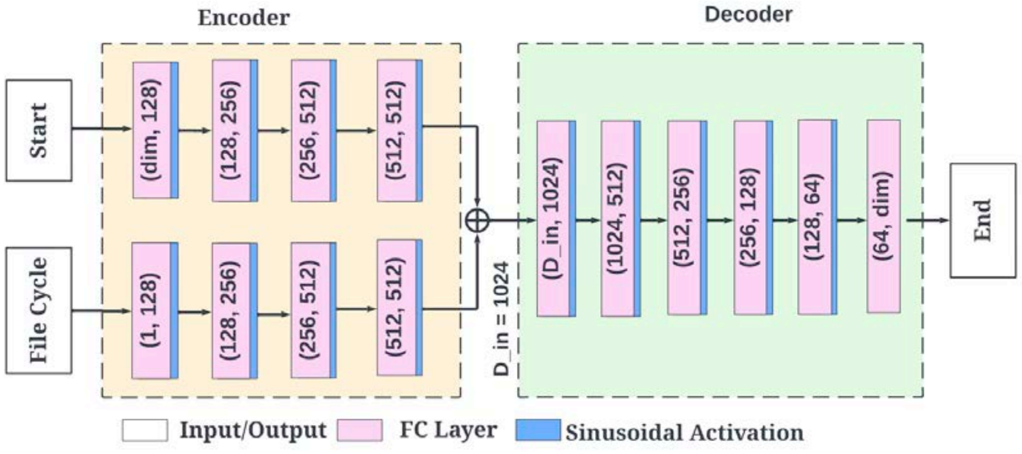
Seeds / Steps	10	50	100	150	200
10	0.26	0.31	0.46	0.41	0.51
100	0.54	0.64	0.73	0.88	0.99
300	0.80	1.03	1.58	1.70	2.04
500	1.09	1.48	1.96	2.43	3.08

Background: Particle Tracing Neural Network Interactive Visualization of Dynamic Flow Fields

- Models trained from Lagrangian-based flow maps:
  - Input: $x_0,\ t\ \rightarrow$ Output: x_t
- Fast inference and small model size that can be deployed on a web-based viewer



The web-based visualization interface, integrated with the particle tracing neural networks, enables users to visualize and explore large 3D time-varying flow fields interactively.

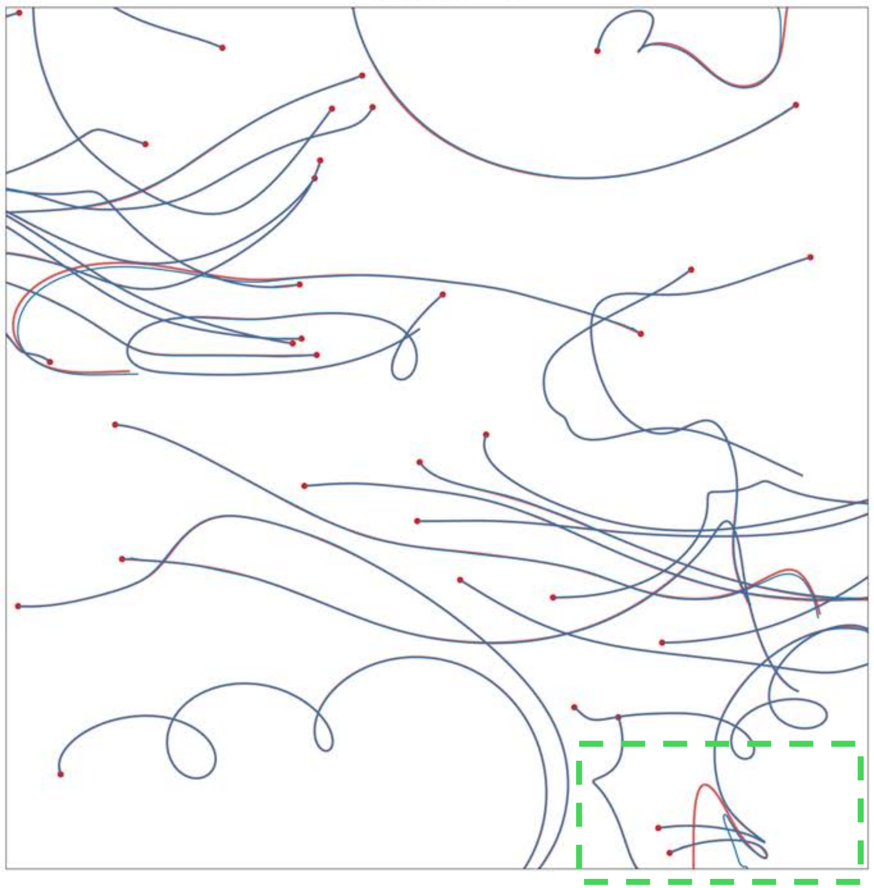


Model Architecture

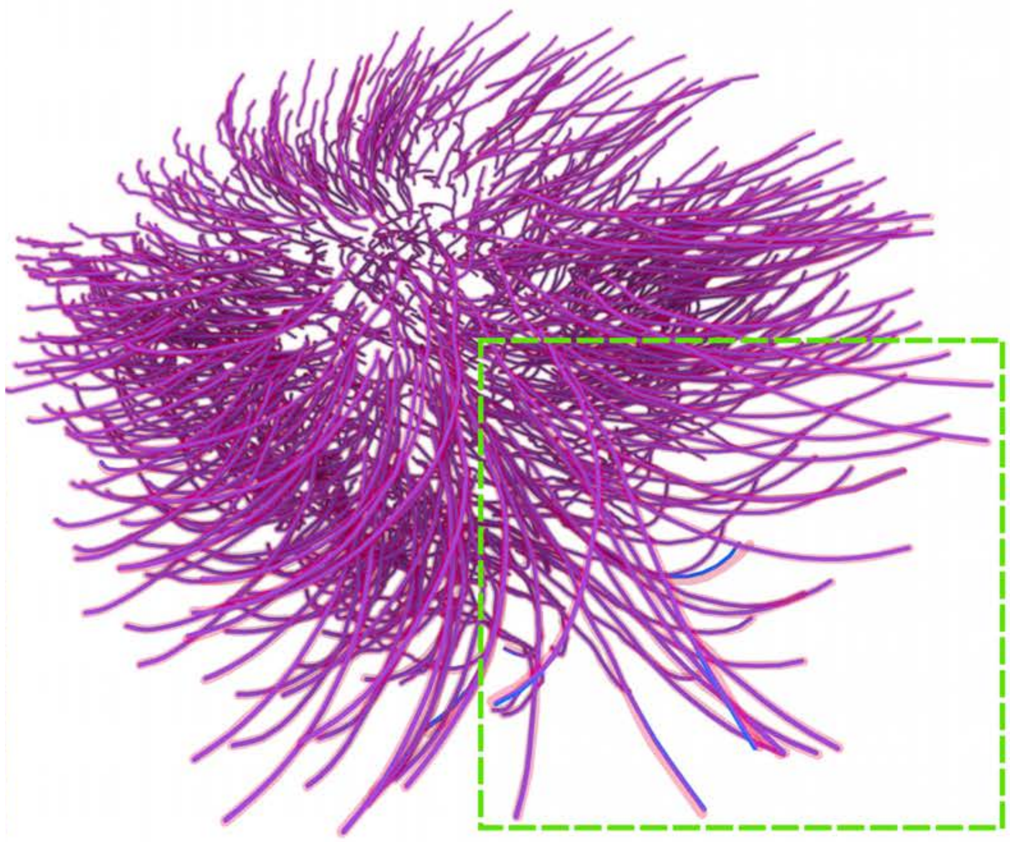
Reference: Han, Mengjiao, et al. "Interactive visualization of time-varying flow fields using particle tracing neural networks." *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*. IEEE, 2024.

Github: https://github.com/MengjiaoH/FlowMap_Web_Viewer

Prediction Errors Are Inevitable



Gerris Flow. The ground truth is colored as red. The predicted pathlines are colored as blue.



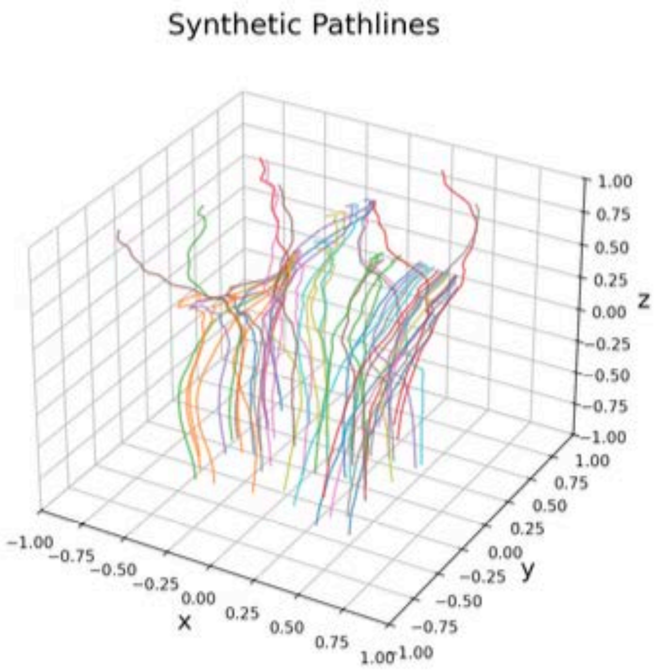
Hurricane Flow. The ground truth is colored as red. The predicted pathlines are colored as blue.

Uncertainty Quantification Methods for Neural Network

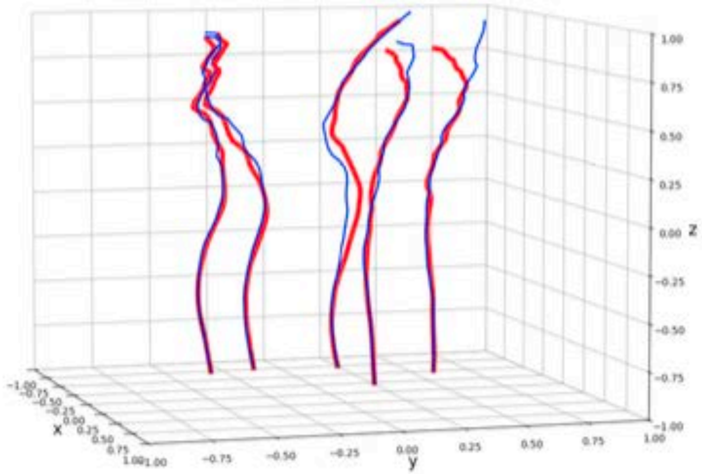
Method	Approach	Strengths	Limitations
Deep Ensembles	Multiple, independent NNs with different random initializations and varied data shuffling orders	High accuracy, reliable uncertainty estimation	High computational cost
Monte Carlo Dropout	Random deactivation of neurons	Computationally efficient, minimal overhead	Approximation may underestimate uncertainty
Stochastic Weight Averaging-Gaussian	Average the network's weights throughout training, then fitting a multivariate Gaussian distribution to these weights	Good balance of efficiency and accuracy	Requires careful hyperparameter tuning

Controlled Experiment with Synthetic Dataset

•Setup: Synthetic flow field with known increasing complexity



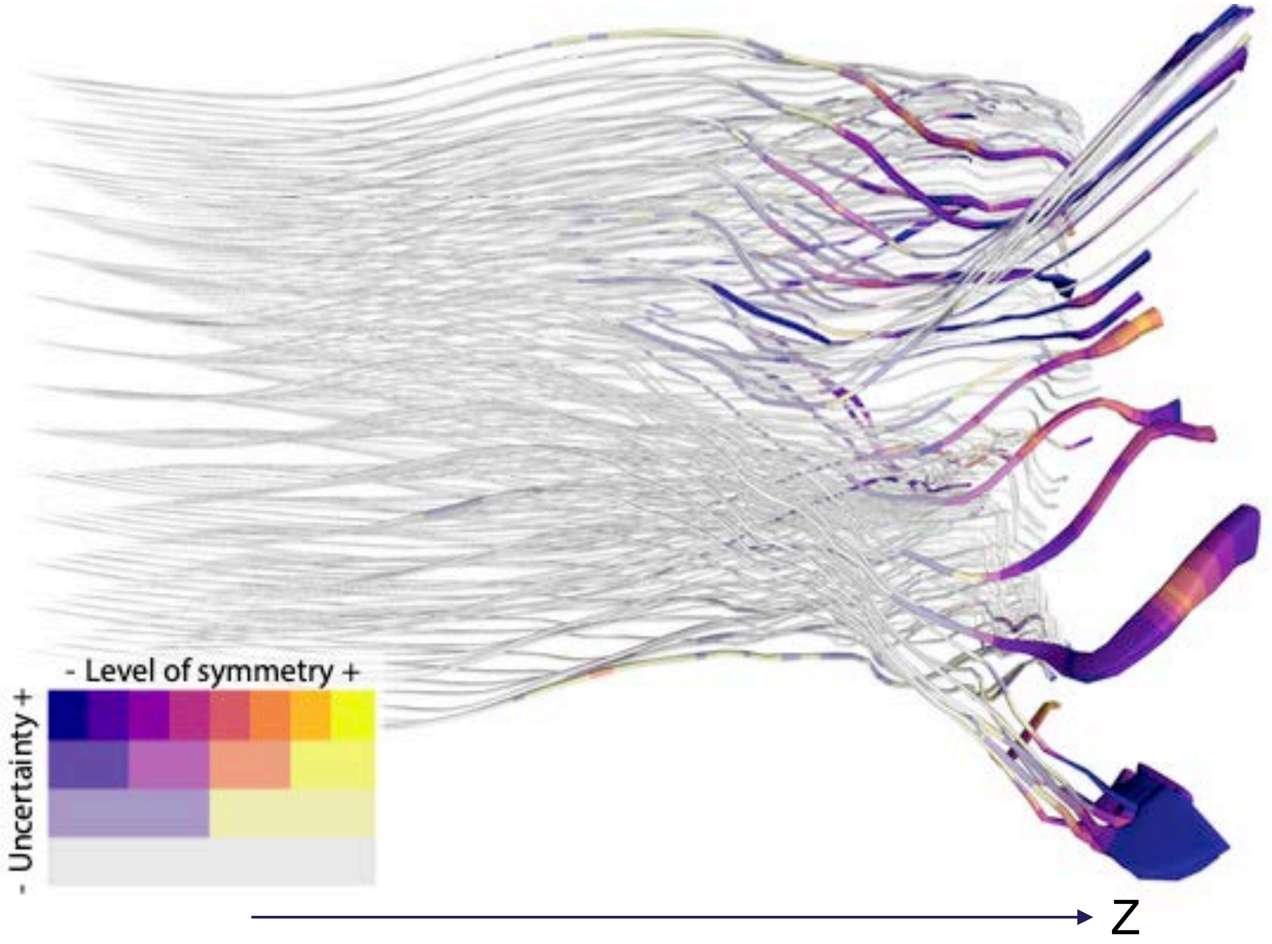
(a) 50 random pathlines



(b) Test data (red) compared to model prediction(blue)

Deep Ensembles

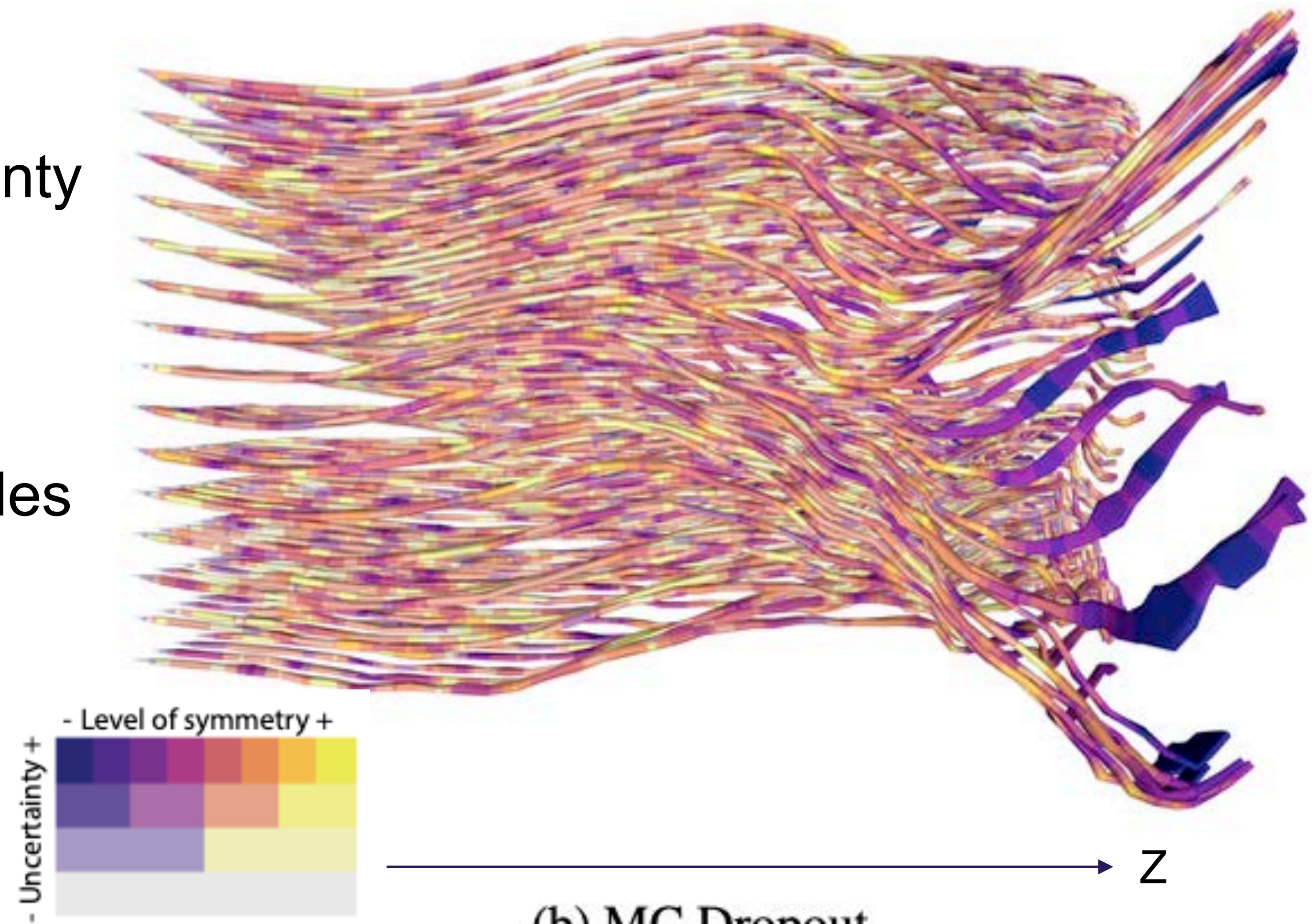
- Deep Ensembles identified **increasing** asymmetric uncertainty
- Results **visually confirmed** error distributions
- 3 hours to train 50 ensembles using two RTX 3090s



(a) Deep Ensembles

Monte Carlo Dropout

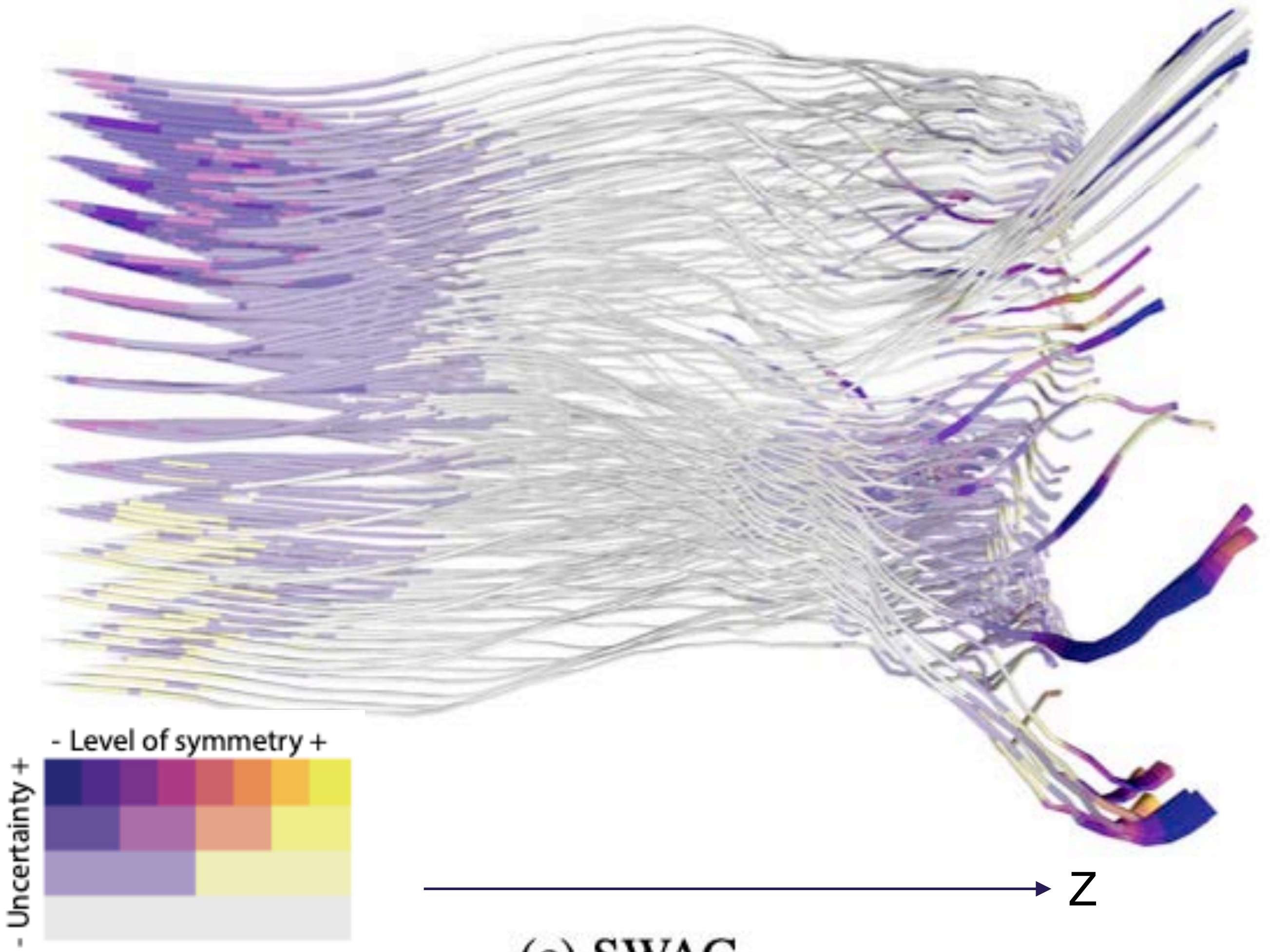
- Slightly **overestimates** low uncertainty regions
- **Underestimates** high uncertainty regions compared to Deep Ensembles
- Less than 0.4 s for 50 ensembles



(b) MC Dropout

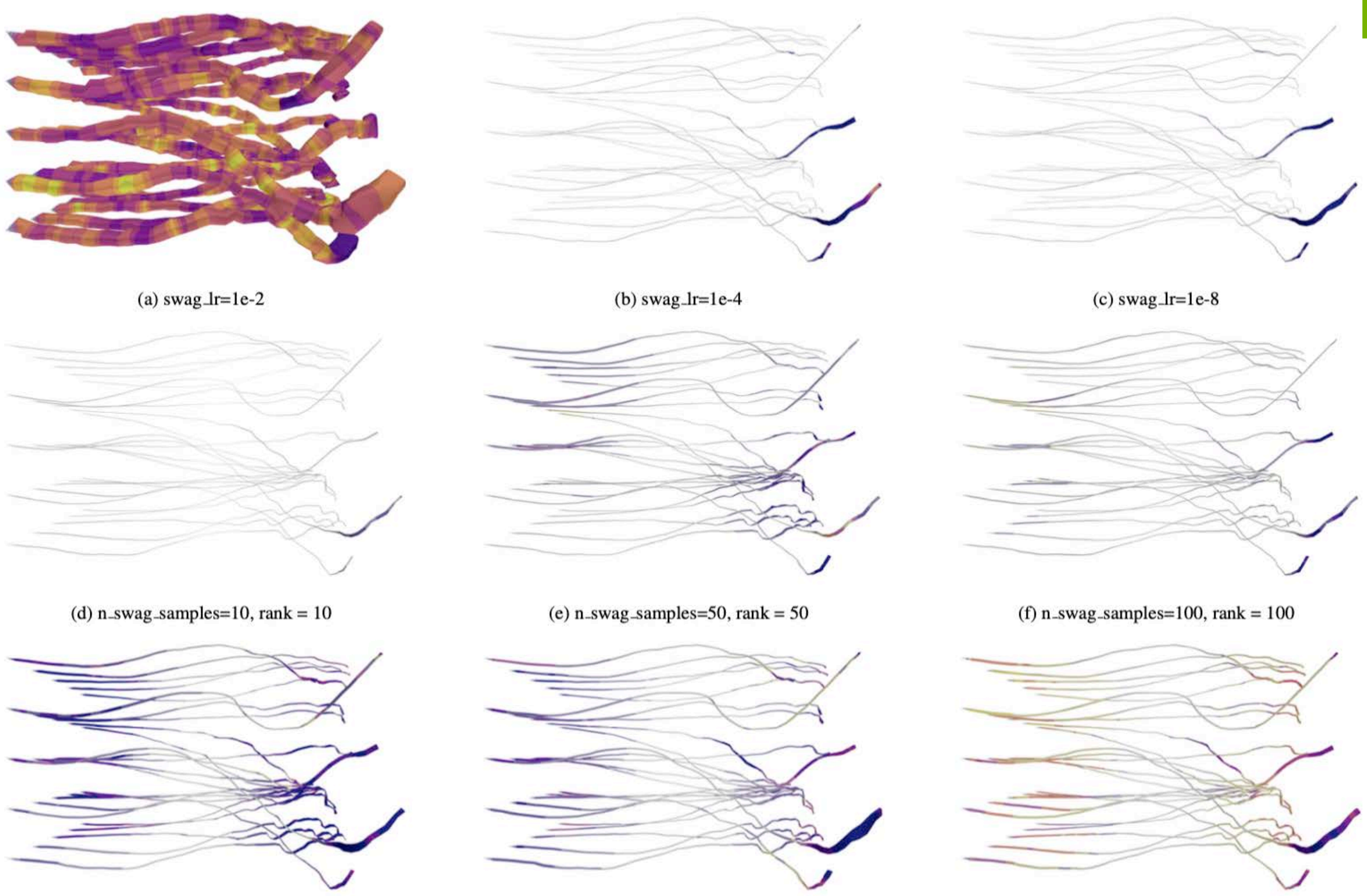
Stochastic Weight Averaging-Gaussian (SWAG)

- Requires **minimal** additional training
- **Hyperparameter tuning** is necessary (learning rate, number of samples, rank)
- 15 s for 50 ensembles using two RTX 3090s
- Good **balance** between accuracy and efficiency



(c) SWAG

SWAG: Hyperparameter Study

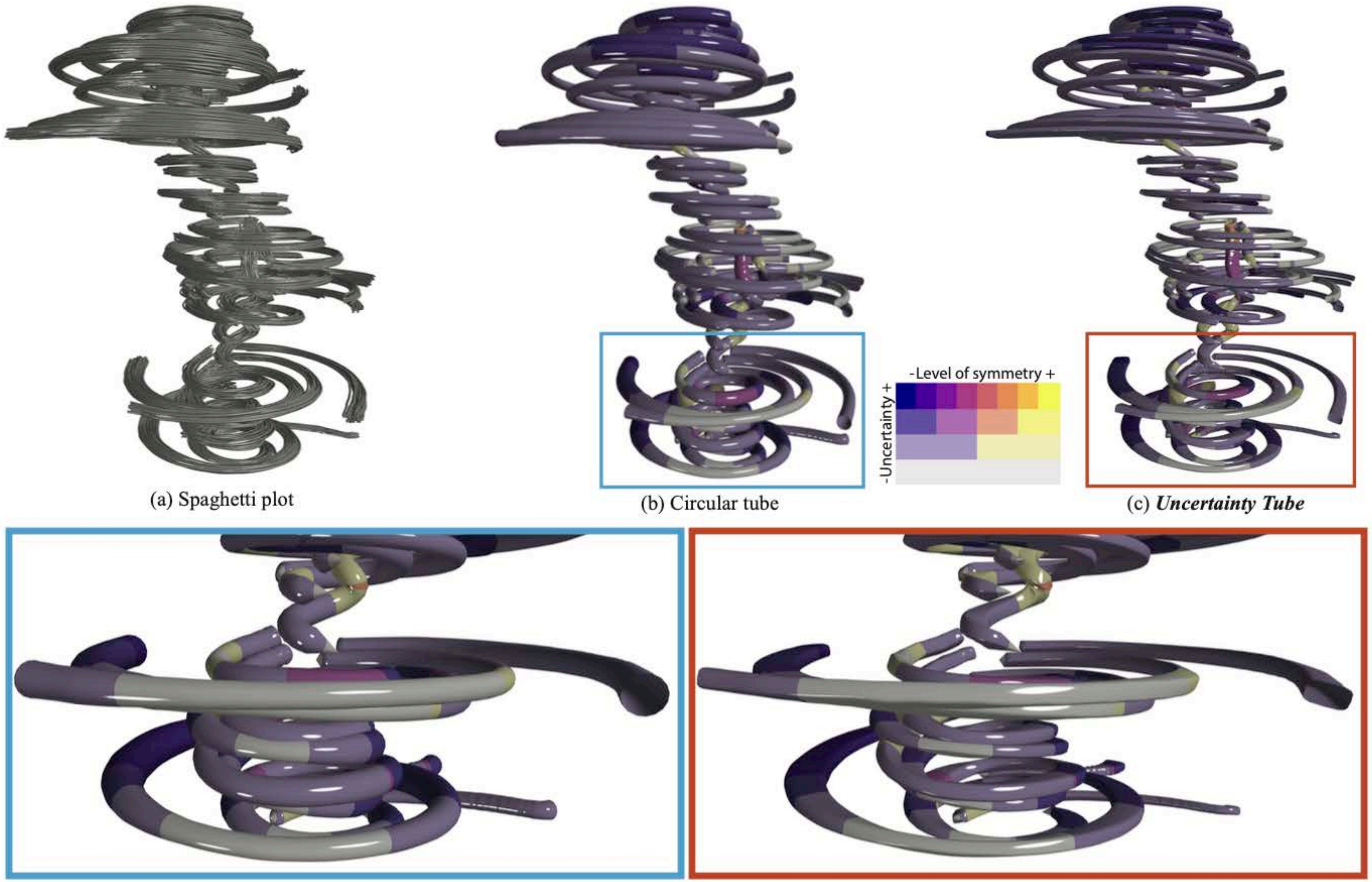


(a) swag_lr=1e-2 (b) swag_lr=1e-4 (c) swag_lr=1e-8

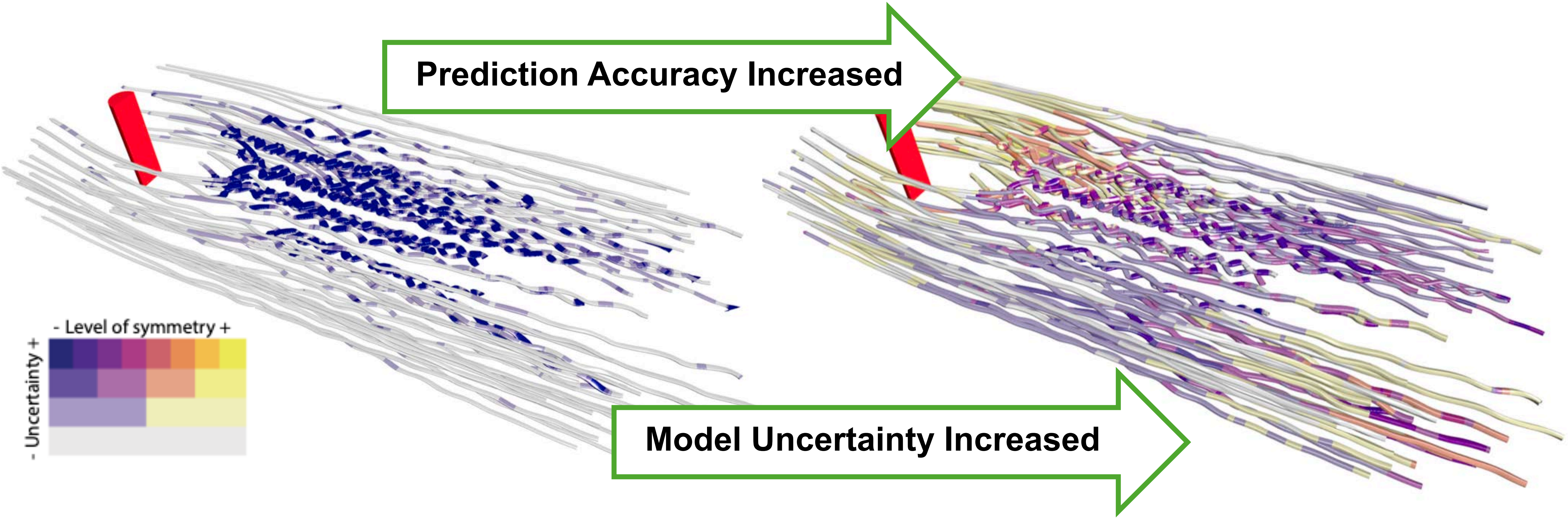
(d) n_swag_samples=10, rank = 10 (e) n_swag_samples=50, rank = 50 (f) n_swag_samples=100, rank = 100

(g) rank=10 (h) rank=50 (i) rank=1000

Tornado Dataset



Uncertainty Estimates Don't Always Match Prediction Errors



Uncertainty visualization from a model trained in Han et al. 2024

Uncertainty visualization from a model trained with spatially uniform scaling

Contributions

- **Introduction of Uncertainty Tube**
  - Utilizes superelliptical tubes to accurately represent asymmetric uncertainty
  - Addresses limitations of conventional symmetric methods (e.g., circular tubes)
  - Represents the direction and evolution of uncertainty along trajectories
  - Improves the interpretability and accuracy of uncertainty visualization
- **Integration of Uncertainty Quantification Methods**
  - Successfully applied with Deep Ensembles, MC Dropout, and SWAG
  - Provide model confidence of the pathline predictions
- **Enhanced Visual Encoding**
  - Uses VSUP-inspired color mapping to distinguish uncertainty levels and symmetry
  - Effective in visualizing complex 3D trajectory uncertainties

Future Work

- **Capture Velocity Uncertainty:** Extend the current method to represent uncertainty in velocity, not just position
- **Visual Encoding Enhancements:** Explore advanced representations such as **textures** or **glyphs** for richer, more intuitive uncertainty cues.
- **Advanced Uncertainty Quantification:** Investigate **fully Bayesian neural networks** and other rigorous methods for deeper uncertainty modeling.
- **Error–Uncertainty Relationship:** Analyze the correlation and divergence between prediction error and estimated uncertainty.

ARGONNE TRAINING PROGRAM ON EXTREME-SCALE COMPUTING

Produced by Argonne National Laboratory, a U.S. Department of Energy Laboratory managed by UChicagoArgonne, LLC under contract DE-AC02-06CH11357.

Special thanks to the National Energy Research Scientific Computing Center (NERSC) and Oak Ridge Leadership Computing Facility (OLCF) for the use of their resources during the training event.

Questions

Reference:
[1] Han, Mengjiao, et al. "Accelerated probabilistic marching cubes by deep learning for time-varying scalar ensembles." 2022 IEEE Visualization and Visual Analytics (VIS). IEEE, 2022.
[2] Han, Mengjiao, et al. "Accelerated Depth Computation for Surface Boxplots with Deep Learning." 2024 IEEE Workshop on Uncertainty Visualization: Applications, Techniques, Software, and Decision Frameworks. IEEE, 2024.
[3] Han, Mengjiao, et al. "Interactive visualization of time-varying flow fields using particle tracing neural networks." *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*. IEEE, 2024.

Github:
[1] https://github.com/MengjiaoH/DeepLearning_LCP
[2] https://github.com/MengjiaoH/SurfaceBoxplot_CXX
[3] https://github.com/MengjiaoH/FlowMap_Web_Viewer