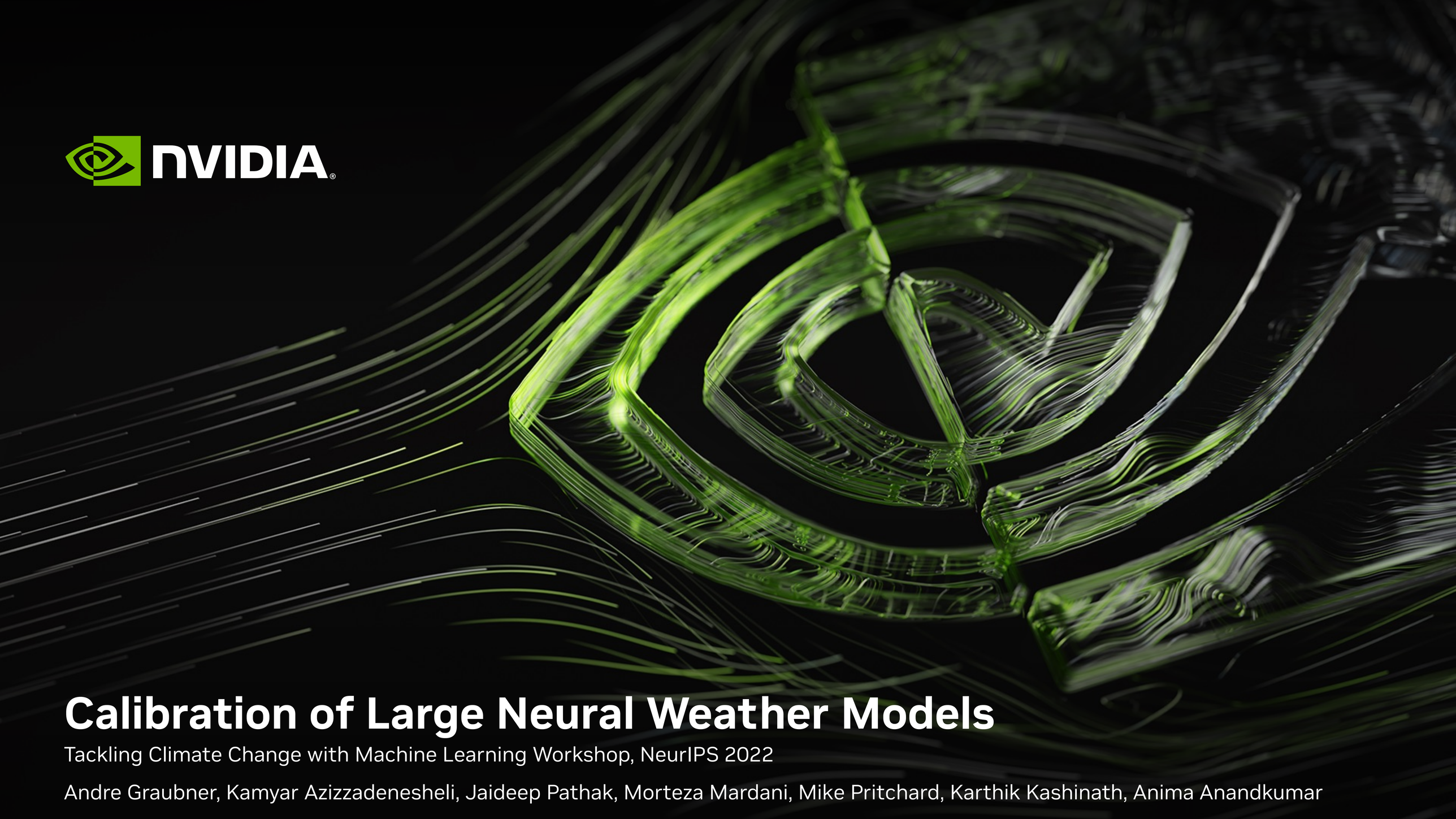


Calibration of Large Neural Weather Models

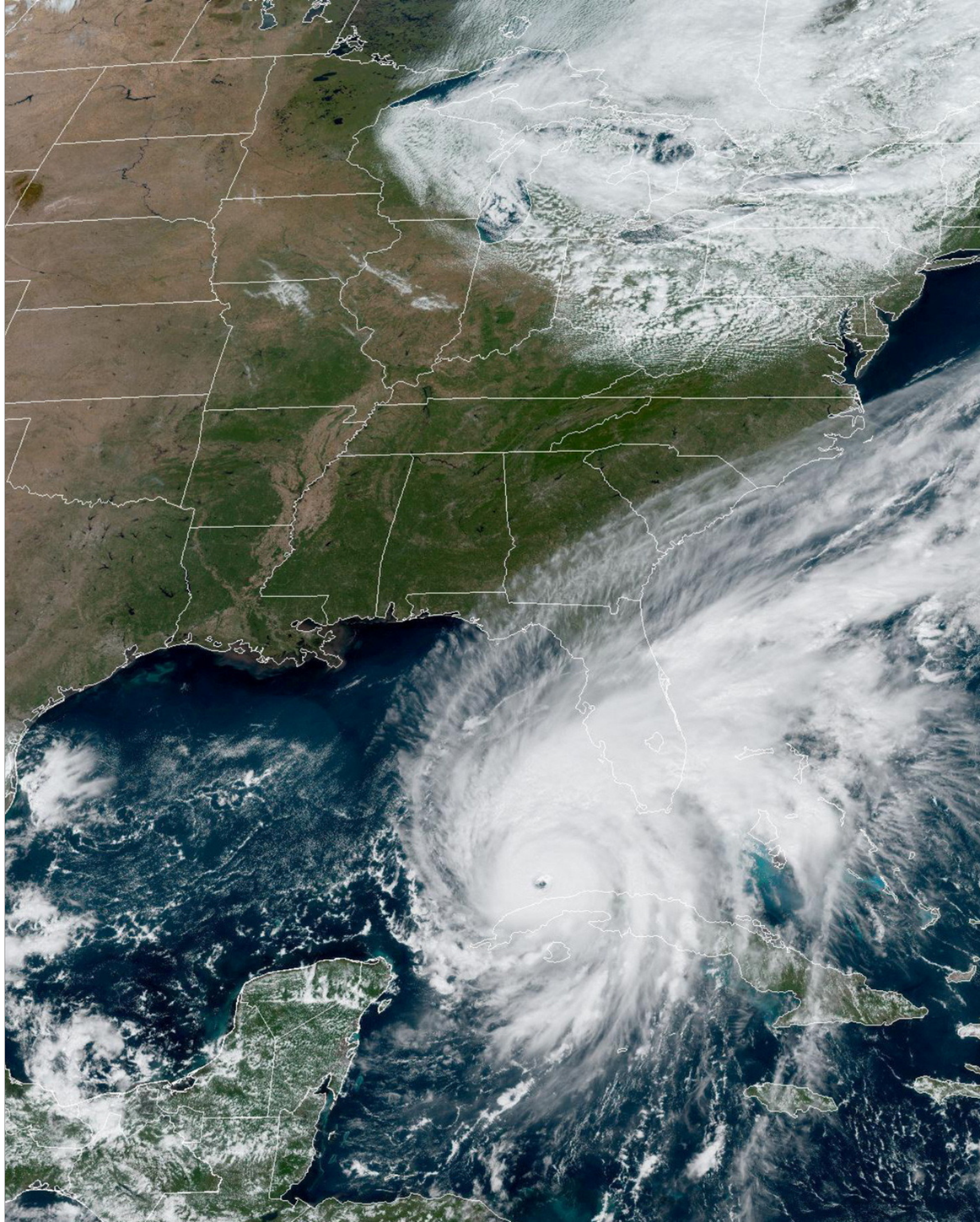
Tackling Climate Change with Machine Learning Workshop, NeurIPS 2022

Andre Graubner, Kamyar Azizzadenesheli, Jaideep Pathak, Morteza Mardani, Mike Pritchard, Karthik Kashinath, Anima Anandkumar

Introduction

- Fast and reliable weather prediction is crucial for mitigating the impact of extreme weather events
- Recently, neural network-based weather models show promising results
  - Significant speedups over classical numerical models
  - Competitive prediction skill at short lead times

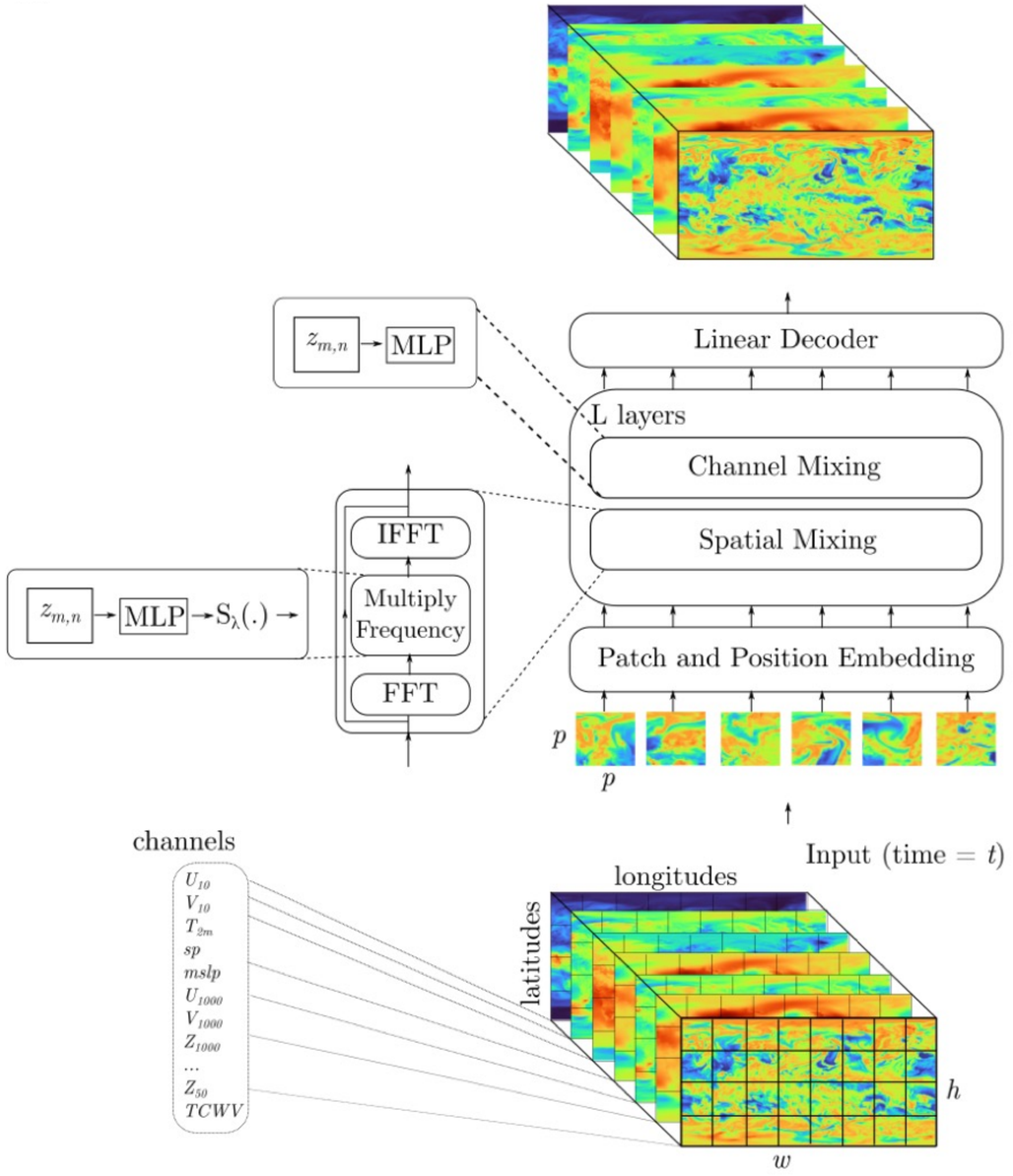
- **Can we produce good uncertainty estimates using such neural weather models?**



FourCastNet

Nvidia's deep learning-based weather model

- Goal: Given atmospheric state, predict atmospheric state 6 hours into the future
- Auto-regressive neural network
- Transformer-based backbone
  - Adaptive Fourier Neural Operator
- Trained on high-resolution (0.25°) ERA5 reanalysis data
- Achieves state-of-the-art performance and scalability at ~45000x speedup against traditional numerical models
- For details, see Pathak et al, 2022

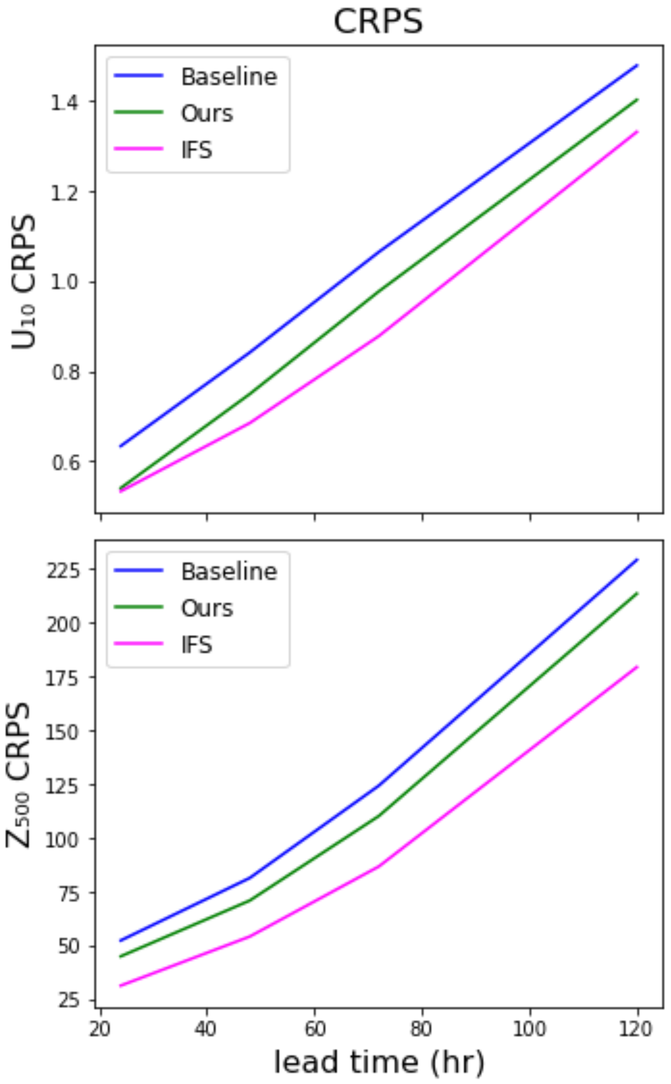


Ensemble Forecasts

- How do we generate forecasts that contain information about predictive uncertainty?
  - Produce **ensemble forecasts** – do not just predict one trajectory, but run N predictions
  - Incorporate uncertainty about initial conditions and about the model
- Many heuristics for producing desired effects
  - Singular vectors, breeding vectors for identifying fast-growing perturbations
  - Perturbations to the physical model based on prior, domain-specific information
- But it is hard to incorporate such hand-crafted heuristics into a neural network-based forecasting model
  - No disentangled representation of individual physical processes
- Can we design ensembling strategies that enable FourCastNet to produce well-calibrated ensembles?

Ensemble Forecasts

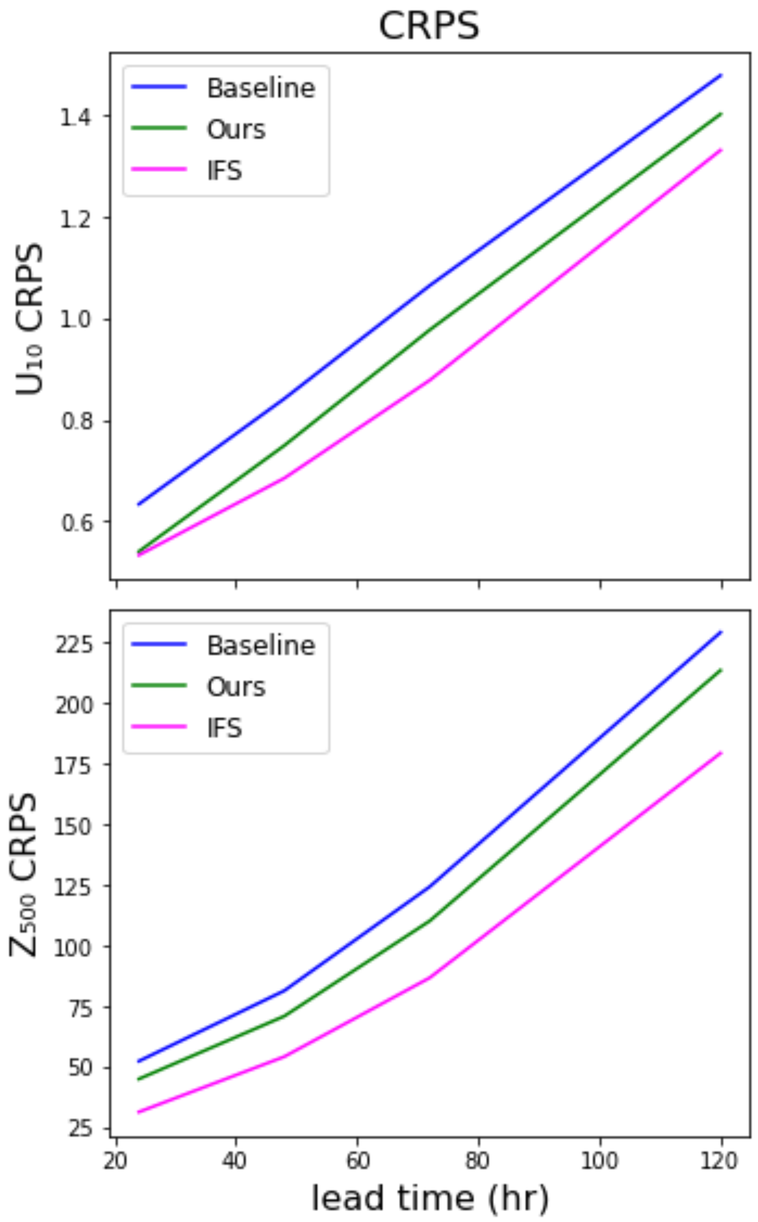
- How do we generate forecasts that contain information about predictive uncertainty?
  - Produce **ensemble forecasts** – do not just predict one trajectory, but run N predictions
  - Incorporate uncertainty about initial conditions and about the model
- Many heuristics for producing desired effects
  - Singular vectors, breeding vectors for identifying fast-growing perturbations
  - Perturbations to the physical model based on prior, domain-specific information
- But it is hard to incorporate such hand-crafted heuristics into a neural network-based forecasting model
  - No disentangled representation of individual physical processes
- Can we design ensembling strategies that enable FourCastNet to produce well-calibrated ensembles?
  - Yes!



Initial Condition Uncertainty

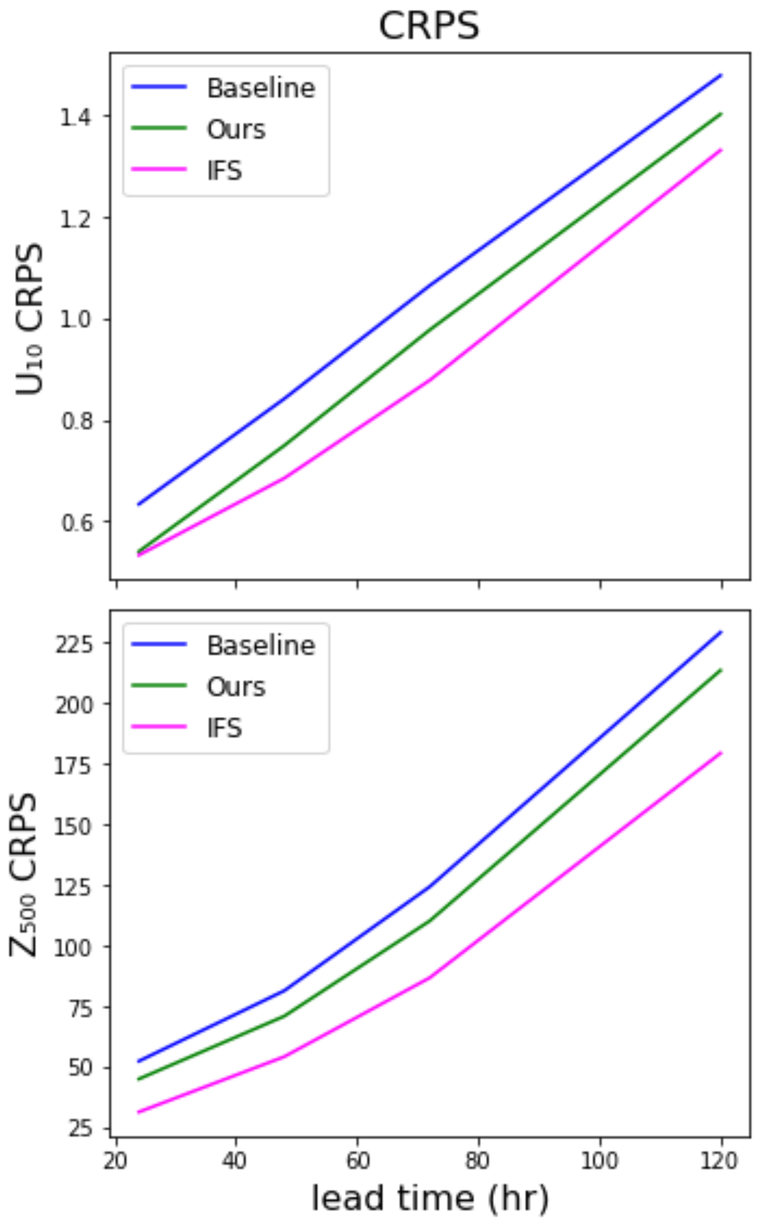
Baseline: Gaussian Perturbations

- Perturb initial condition using uncorrelated Gaussian noise



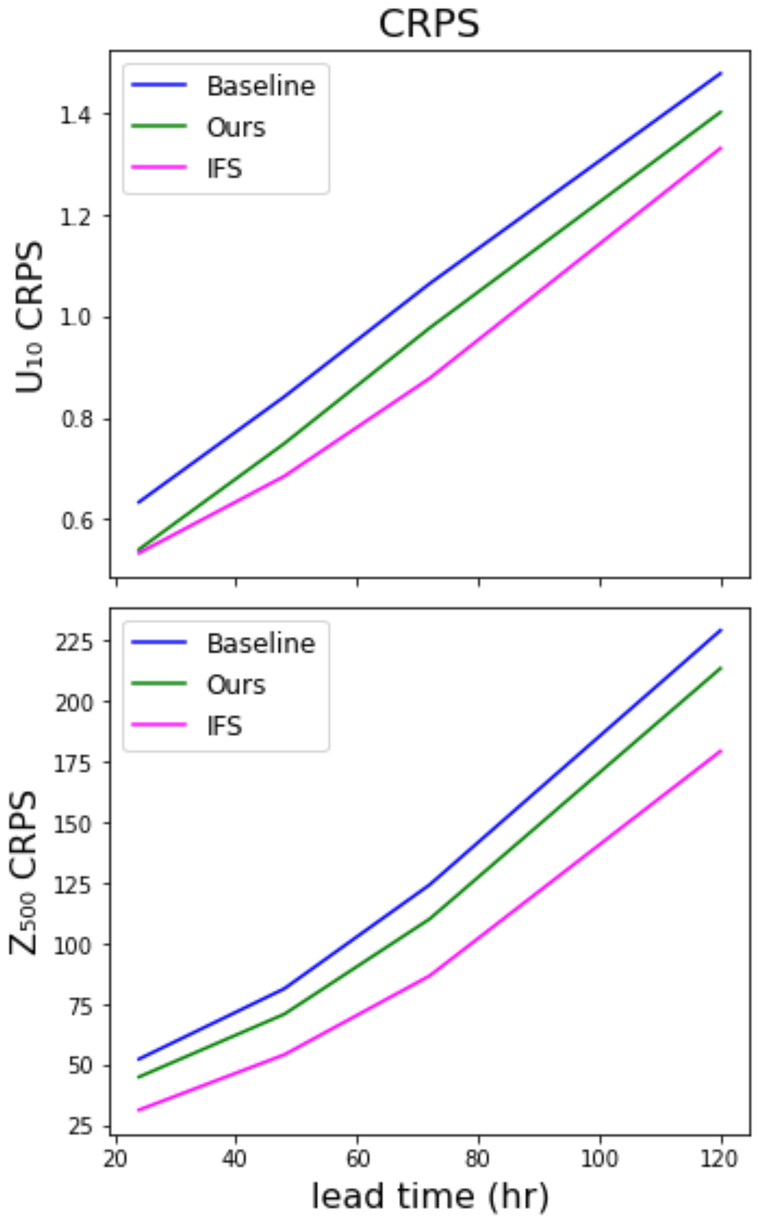
Baseline: Gaussian Perturbations

- Perturb initial condition using uncorrelated Gaussian noise
- **Hypothesis**: FourCastNet acts like a physical system, being more sensitive to perturbations at larger length-scales
  - Uncorrelated Gaussian noise quickly disperses and the ensemble collapses



Correlated Perturbations

- To investigate our hypothesis, we add spatial correlation to the initial condition perturbations

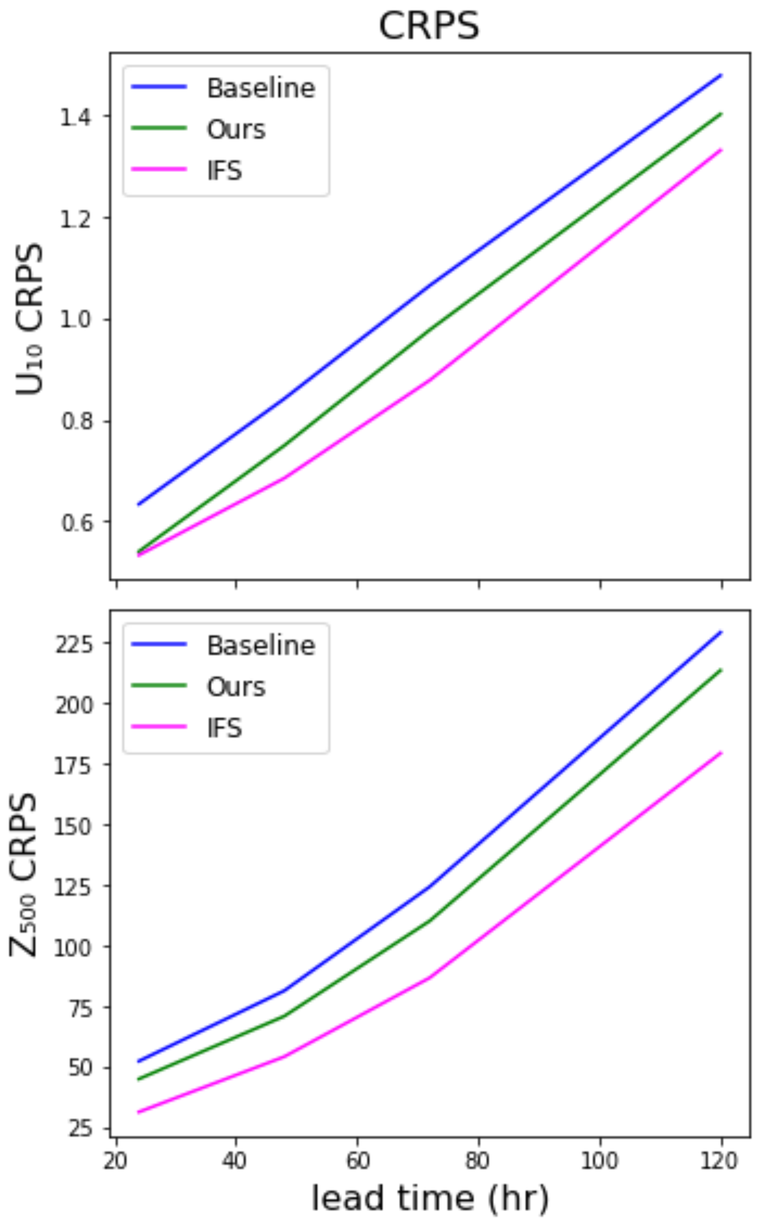


Correlated Perturbations

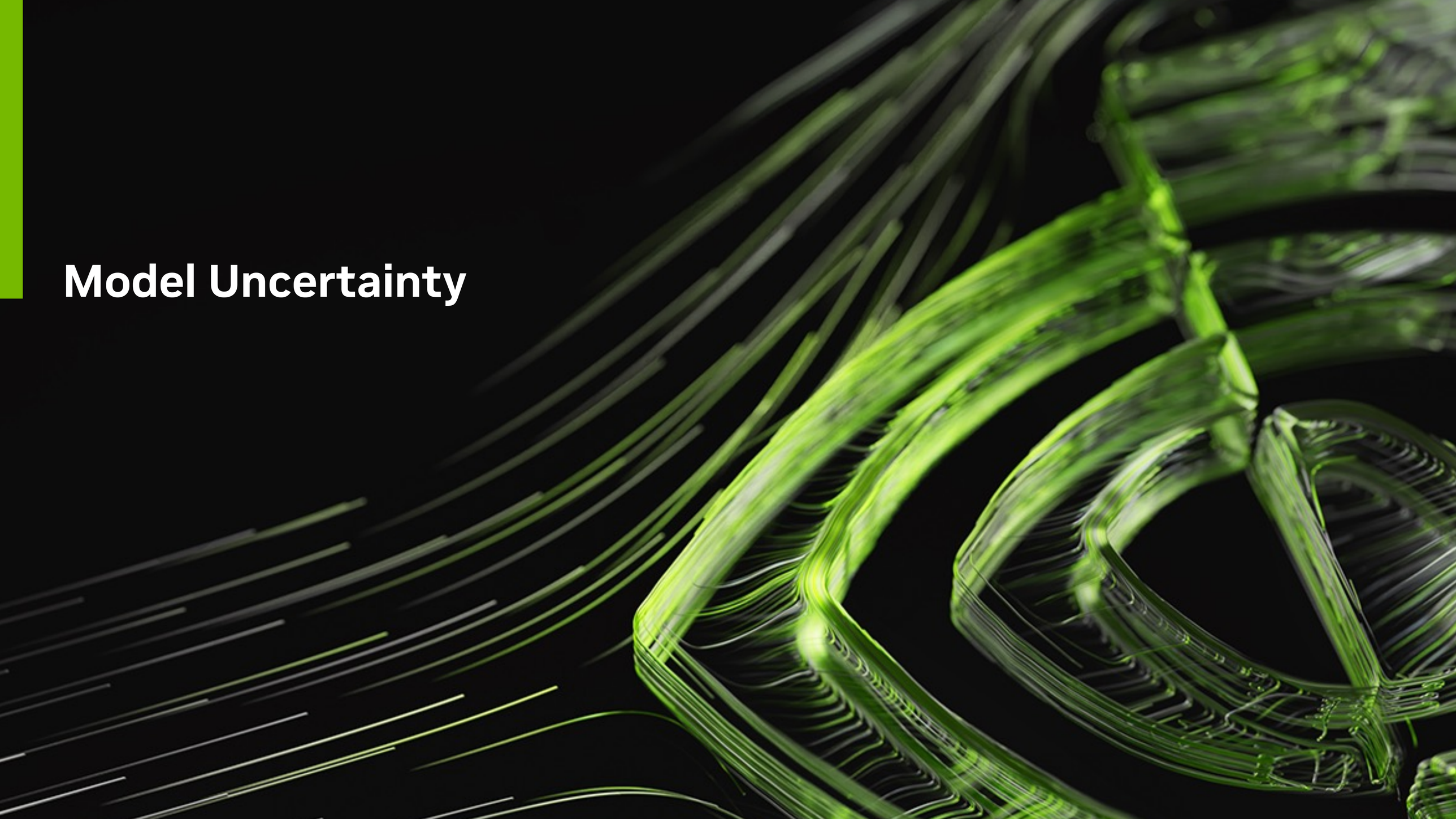
- To investigate our hypothesis, we add spatial correlation to the initial condition perturbations
  - Sample uncorrelated gaussian noise
  - Transform to frequency domain
  - Rescale frequencies in 2d frequency domain to be proportional to

$$\frac{1}{f_x + f_y}$$

  - Transform back to spatial domain and apply perturbation



Model Uncertainty

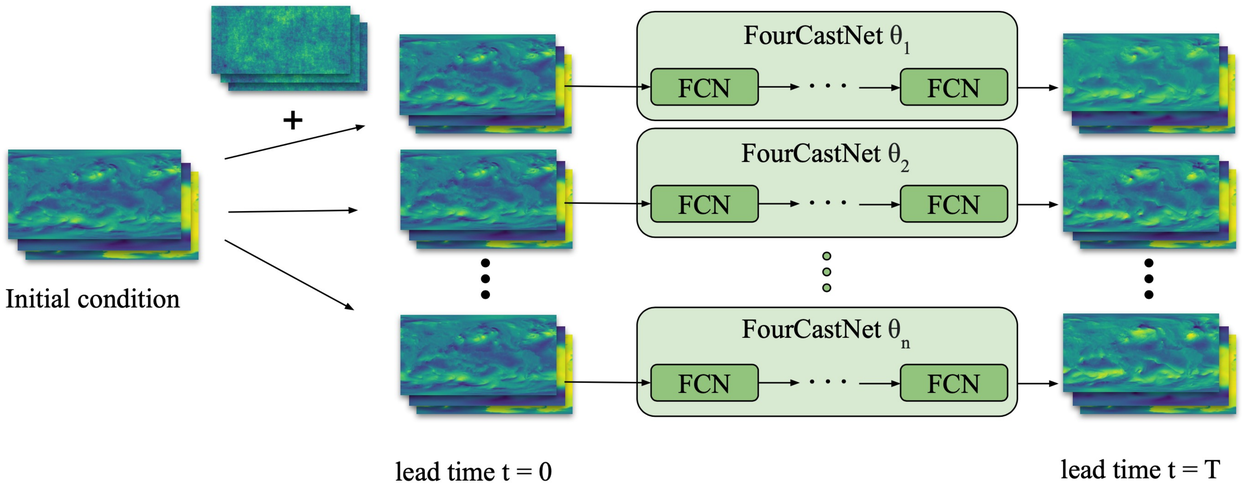


Initial Condition Uncertainty is not enough

- Numerical models routinely include perturbations to the **weather model itself**, to capture uncertainty arising from the parametrisation of the model instead of the initial condition
- How to incorporate this intuition into a deep learning based setup?

Initial Condition Uncertainty is not enough

- Numerical models routinely include perturbations to the **weather model itself**, to capture uncertainty arising from the parametrisation of the model instead of the initial condition
- How to incorporate this intuition into a deep learning based setup?
  - Can not use heuristcs for perturbing parametrisations of physical processes based on prior knowledge directly!
  - Use methods from Bayesian deep learning to quantify uncertainty over weights, then sample different models during inference



Stochastic Weight Averaging - Gaussian

- Learn a posterior distribution over neural network weights given training data
  - During inference, sample N different models from this posterior distribution and use them to process different trajectories
  - How to approximate this posterior?

Stochastic Weight Averaging - Gaussian

- Learn a posterior distribution over neural network weights given training data
  - During inference, sample N different models from this posterior distribution and use them to process different trajectories
  - How to approximate this posterior?
- SWA-G: Approximate posterior as a Gaussian with a combination of low-rank and diagonal covariance
- Train model to convergence using regular methods
- Keep training using SGD with a constant learning rate and periodically save weight checkpoints

$$\mathcal{N}(\theta_{\text{SWA}}, \frac{1}{2} \cdot (\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}))$$

Stochastic Weight Averaging - Gaussian

- Learn a posterior distribution over neural network weights given training data
  - During inference, sample N different models from this posterior distribution and use them to process different trajectories
  - How to approximate this posterior?
- SWA-G: Approximate posterior as a Gaussian with a combination of low-rank and diagonal covariance
- Train model to convergence using regular methods
- Keep training using SGD with a constant learning rate and periodically save weight checkpoints

$$\mathcal{N}(\theta_{\text{SWA}}, \frac{1}{2} \cdot (\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}))$$

Mean of model checkpoints

Stochastic Weight Averaging - Gaussian

- Learn a posterior distribution over neural network weights given training data
  - During inference, sample N different models from this posterior distribution and use them to process different trajectories
  - How to approximate this posterior?
- SWA-G: Approximate posterior as a Gaussian with a combination of low-rank and diagonal covariance
- Train model to convergence using regular methods
- Keep training using SGD with a constant learning rate and periodically save weight checkpoints

$$\mathcal{N}(\theta_{\text{SWA}}, \frac{1}{2} \cdot (\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}))$$

Per-weight variance in the checkpoints

Stochastic Weight Averaging - Gaussian

- Learn a posterior distribution over neural network weights given training data
  - During inference, sample N different models from this posterior distribution and use them to process different trajectories
  - How to approximate this posterior?
- SWA-G: Approximate posterior as a Gaussian with a combination of low-rank and diagonal covariance
- Train model to convergence using regular methods
- Keep training using SGD with a constant learning rate and periodically save weight checkpoints

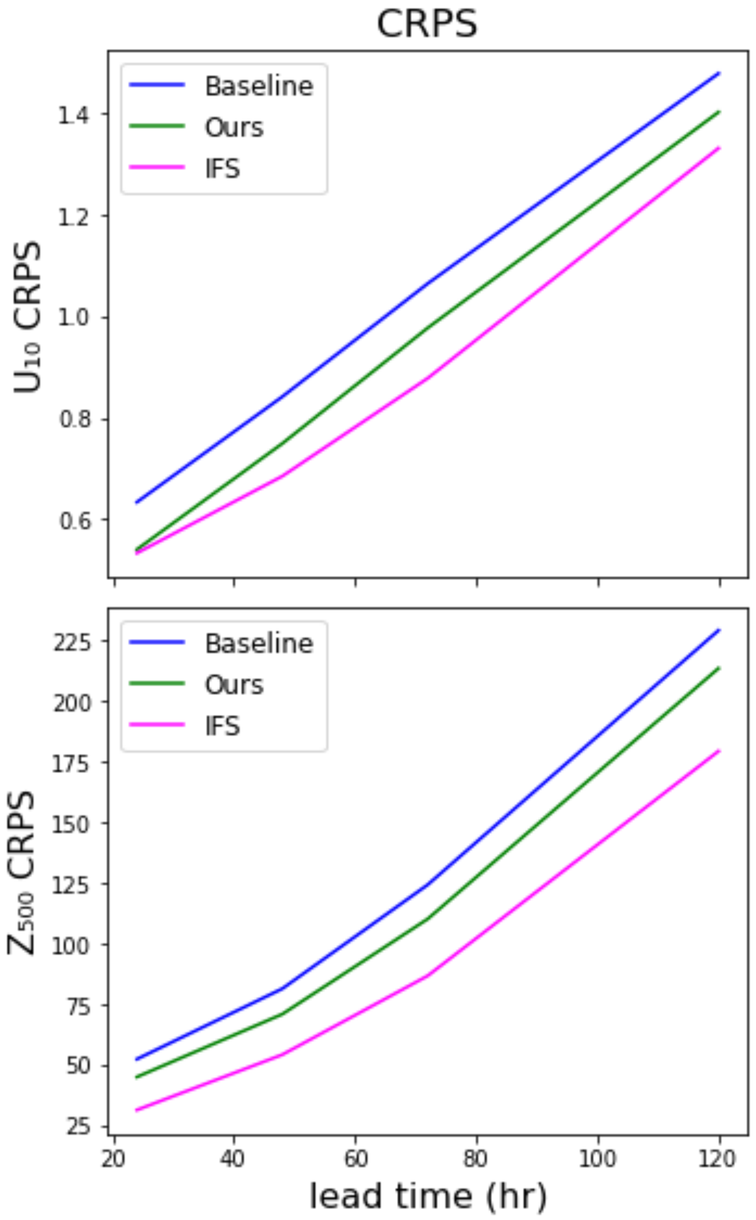
$$\mathcal{N}(\theta_{\text{SWA}}, \frac{1}{2} \cdot (\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}))$$

$$\Sigma_{\text{low-rank}} = \frac{1}{K-1} \cdot \hat{D}\hat{D}^T$$

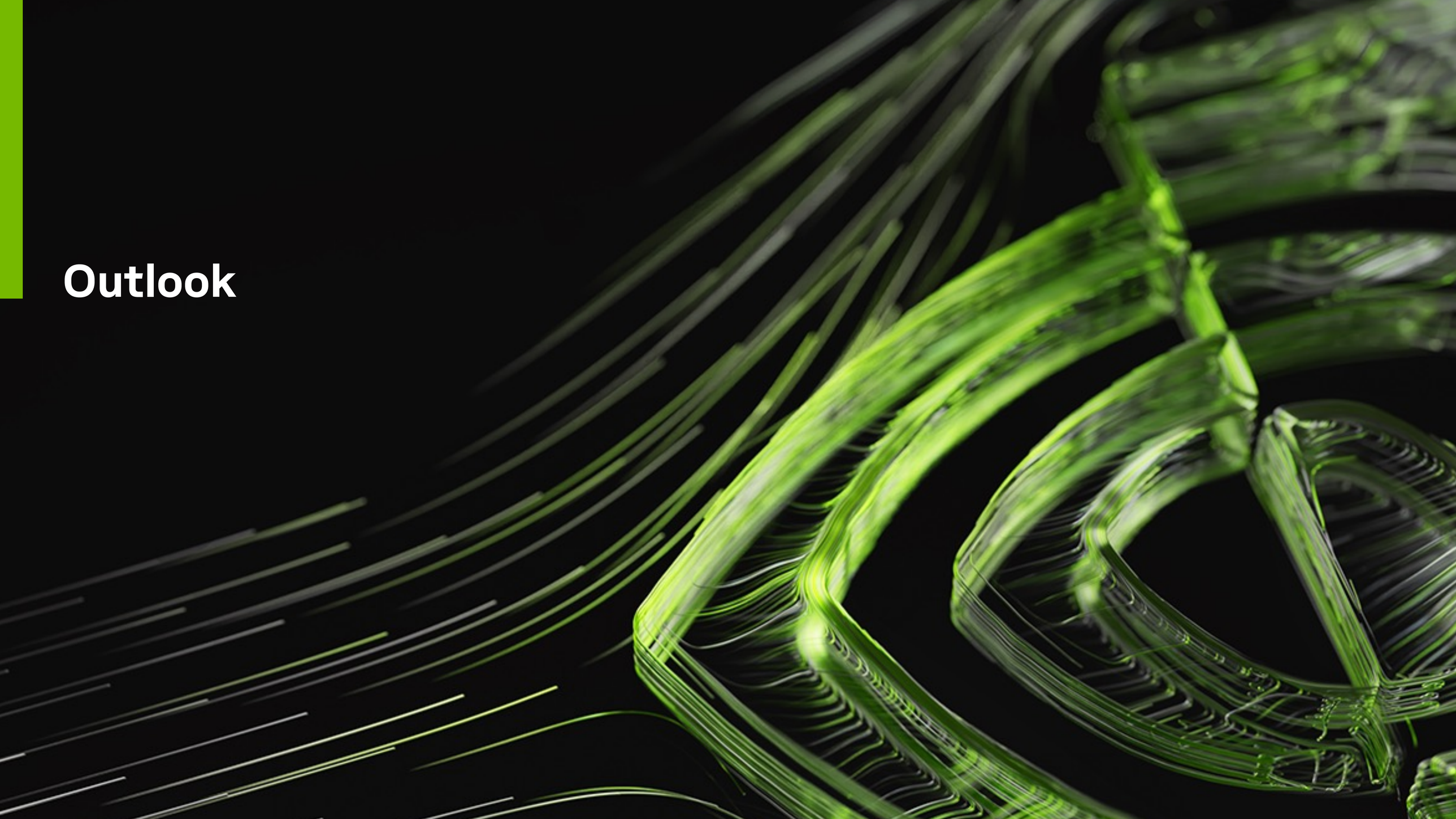
Matrix with columns corresponding to deviations of last K checkpoints to from the sample mean

Stochastic Weight Averaging - Gaussian

- Learn a posterior distribution over neural network weights given training data
  - During inference, sample N different models from this posterior distribution and use them to process different trajectories
  - How to approximate this posterior?
- SWA-G: Approximate posterior as a Gaussian with a combination of low-rank and diagonal covariance
- Train model to convergence using regular methods
- Keep training using SGD with a constant learning rate and periodically save weight checkpoints



Outlook



Possible Directions for Future Work

- Apply classical methods from numerical weather prediction
  - Singular vectors, breeding vectors
  - Comes with its own set of challenges, not clear that this is exactly what we want…

Possible Directions for Future Work

- Apply classical methods from numerical weather prediction
  - Singular vectors, breeding vectors
  - Comes with its own set of challenges, not clear that this is exactly what we want…

- Actively lean into the generative modeling formulation (e.g. using diffusion models)
  - Can this scale to large domains like ours, or even larger domains in future high-resolution models?
  - In principle, sure. But far from trivial to get this right!

Possible Directions for Future Work

- Apply classical methods from numerical weather prediction
  - Singular vectors, breeding vectors
  - Comes with its own set of challenges, not clear that this is exactly what we want…
- Actively lean into the generative modeling formulation (e.g. using diffusion models)
  - Can this scale to large domains like ours, or even larger domains in future high-resolution models?
  - In principle, sure. But far from trivial to get this right!
- Improve the Bayesian approach
  - Applying more expressive methods for approximating the posterior might be required
  - Even if SWA-G is enough, there might be heuristics that can be applied to use the information about the posterior to sample better ensembles of models

Possible Directions for Future Work

- Apply classical methods from numerical weather prediction
  - Singular vectors, breeding vectors
  - Comes with its own set of challenges, not clear that this is exactly what we want…
- Actively lean into the generative modeling formulation (e.g. using diffusion models)
  - Can this scale to large domains like ours, or even larger domains in future high-resolution models?
  - In principle, sure. But far from trivial to get this right!
- Improve the Bayesian approach
  - Applying more expressive methods for approximating the posterior might be required
  - Even if SWA-G is enough, there might be heuristics that can be applied to use the information about the posterior to sample better ensembles of models
- Thank you for your attention!