MACHINE LEARNING
Science and Technology

PURPOSE-LED PUBLISHING™ Part of the MACHINE LEARNING Series

PAPER • OPEN ACCESS

Uncertainty quantification of surrogate models using conformal prediction

To cite this article: Vignesh Gopakumar *et al* 2026 *Mach. Learn.: Sci. Technol.* **7** 015025

View the article online for updates and enhancements.

You may also like

- Suppressing simulation bias in multi-modal data using transfer learning
Bogdan Kustowski, Jim A Gaffney, Brian K Spears et al.
- Parameter uncertainties for imperfect surrogate models in the low-noise regime
Thomas D Swinburne and Danny Perez
- Robust and scalable uncertainty estimation with conformal prediction for machine-learned interatomic potentials
Yuge Hu, Joseph Musielewicz, Zachary W Ulissi et al.

This content was downloaded from IP address 195.176.113.76 on 25/03/2026 at 07:45

MACHINE LEARNING
Science and Technology

Check for updates

OPEN ACCESS

RECEIVED
11 August 2025

REVISED
26 November 2025

ACCEPTED FOR PUBLICATION
17 December 2025

PUBLISHED
3 February 2026

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Uncertainty quantification of surrogate models using conformal prediction

Vignesh Gopakumar[1,5,*], **Ander Gray**[2,5], **Joel Oskarsson**[3], **Lorenzo Zanisi**[4], **Daniel Giles**[1], **Matt J Kusner**[1], **Stanislas Pamela**[4] and **Marc Peter Deisenroth**[1]

[1] Centre for Artificial Intelligence, Department of Computer Science, University College London, London WC1V 6LJ, United Kingdom
[2] Heudiasyc Laboratory, Universitè de Technologie de Compiègne, Compiègne 60200, France
[3] Department of Computer and Information Science, Linköping University, Linköping 581 83, Sweden
[4] Computing Division, UK Atomic Energy Authority, Oxford OX14 3EB, United Kingdom
[5] These authors contributed equally to this work.
[*] Author to whom any correspondence should be addressed.

E-mail: v.gopakumar@ucl.ac.uk, vignesh.gopakumar@ukaea.uk, ander.gray@hds.utc.fr, joel.oskarsson@outlook.com, lorenzo.zanisi@ukaea.uk, d.giles@ucl.ac.uk, m.kusner@ucl.ac.uk, stanislas.pamela@ukaea.uk and m.deisenroth@ucl.ac.uk

Keywords: surrogate models, uncertainty quantification, conformal prediction, neural-PDE, neural-weather

Abstract

Data-driven surrogate models offer fast, inexpensive approximations to complex numerical and experimental systems but typically lack uncertainty quantification, limiting their reliability in safety-critical applications. While Bayesian methods provide uncertainty estimates, they offer no statistical guarantees and struggle with high-dimensional spatio-temporal problems due to computational costs and dependence on prior specification. We present a conformal prediction (CP) framework that provides statistically guaranteed marginal coverage for surrogate models in a model-agnostic manner with near-zero computational cost. Our approach handles high-dimensional spatio-temporal outputs by performing cell-wise calibration while preserving the tensorial structure of predictions. Through extensive empirical evaluation across diverse applications-including partial differential equations, magnetohydrodynamics, weather forecasting, and fusion diagnostics-we demonstrate that CP achieves empirical coverage with valid error bars regardless of model architecture (Multi-layer perceptrons, U-Net, Fourier neural operator, ViT, GNN), training regime, or output dimensionality (spanning 32 to over 20 million dimensions). We evaluate three nonconformity scores (conformalised quantile regression, absolute error residual, and standard deviation) for both deterministic and probabilistic models, showing that guaranteed coverage holds even for out-of-distribution predictions where models are deployed on physics regimes different from their training data. Calibration requires only seconds to minutes on standard hardware, with prediction set construction incurring negligible computational overhead. The framework enables rigorous validation of pre-trained surrogate models for downstream applications without retraining, providing actionable uncertainty quantification for decision-making in scientific domains. While CP provides marginal rather than conditional coverage and assumes exchangeability between calibration and test data-limitations we demonstrate empirically through sensitivity analyses-our method circumvents the curse of dimensionality inherent in traditional uncertainty quantification approaches, offering a practical tool for the trustworthy deployment of machine learning in the physical sciences.

1. Introduction

Partial differential equations (PDEs) governing physical processes are solved using complex numerical simulation codes. While these codes offer mathematically rigorous solutions, they are limited to discretised domains and require computationally expensive iterative solvers such as finite-volume and finite-element methods. Such simulation codes are central to scientific disciplines in biology (Hospital

© 2026 The Author(s). Published by IOP Publishing Ltd

et al 2015), engineering (Giudicelli *et al* 2024), and climate science (Danabasoglu *et al* 2020, Lavin *et al* 2021), but are difficult to deploy for rapid, iterative modelling required while exploring design spaces. Machine learning offers an alternative data-driven route for obtaining quick, inexpensive approximations to numerical simulations (Bertone *et al* 2019, Karniadakis *et al* 2021). Data-driven surrogate models distil spatio-temporal information from simulations into parameterised machine learning models. Due to their efficiency, cost-effectiveness, and relative accuracy, neural networks have become ubiquitous within scientific modelling, with primary importance in tackling large-scale PDEs in climate (Kurth *et al* 2023, Lam *et al* 2023), computational fluid dynamics (CFD) (Jiang *et al* 2020, Pfaff *et al* 2021), and nuclear fusion (Gopakumar and Samaddar 2020, van de Plassche *et al* 2020).

However, these surrogate models remain approximations of true physical systems, inheriting multiple layers of uncertainty from both the numerical codes and the PDE formulations themselves. Critically, they often fail to quantify their approximation error relative to the numerical code, producing overconfident outputs regardless of their training domain. This poses two key problems: (a) without confidence assessment, erroneous predictions can lead to severe downstream consequences; (b) high training costs are wasted if predictions lack actionable uncertainty quantification. While several works have attempted uncertainty estimation for surrogate models (Geneva and Zabaras 2020, Alhajeri *et al* 2022, Psaros *et al* 2023, Zou *et al* 2024), they fail to provide statistical guarantees over error bars and do not scale to complex scenarios (Abdar *et al* 2021). Moreover, they require computationally expensive ensemble training Lakshminarayanan *et al* (2017), extensive sampling (MacKay 1992), or architectural modifications (Gal and Ghahramani 2016). Validating surrogate model outputs for specific downstream applications remains a pressing challenge.

Conformal prediction (CP) (Vovk *et al* 2005) provides a framework for computing statistically guaranteed error bars over pre-trained and fine-tuned models, i.e. error bars calibrated to provide required coverage. CP relies on calibrating model performance across a dataset representative of the desired prediction distribution, then utilising these calibration measures to provide valid error bars for model outputs.

In this paper, we conduct a thorough empirical study demonstrating that CP provides statistically guaranteed error bars for neural-network-based surrogate models across spatio-temporal domains, even in out-of-training-distribution scenarios. Through experiments of increasing complexity, we show guaranteed marginal coverage irrespective of model choice (deterministic or probabilistic), output dimensionality (up to 20 million dimensions), training data, or physical setting. We explore various CP methods, comparing their cost, performance, and architectural requirements. Our work provides a rigorous method to assess the usefulness, validity, and applicability of pre-trained and fine-tuned surrogate models for inference and production scenarios.

Applications to critical scientific challenges

Machine-learning-based surrogate modelling accelerates scientific simulation through computational efficiency and enables data-driven discovery at scale. When modelling complex systems such as CFDs, nuclear fusion and weather forecasting, both computational speed and accurate uncertainty estimates are crucial. In safety-critical systems Knight (2002), supplementing model predictions with calibrated uncertainty estimations is imperative for improved decision-making in downstream tasks. The CP framework demonstrated here advances uncertainty quantification for complex scientific modelling at scale with industry-level safety-critical applications. Neural-PDE solvers offer quick, inexpensive PDE solutions (Yin *et al* 2023), enabling system understanding and optimal design point identification (Li *et al* 2023, Shukla *et al* 2024). As these models become ubiquitous, verifying prediction accuracy becomes pressingly important. Applications span nuclear fusion for low-carbon energy production (Kates-Harbeck *et al* 2019, Linke *et al* 2019, Degrave *et al* 2022, Lerede *et al* 2023, Carey *et al* 2024, Pamela *et al* 2024) to weather forecasting for proactive climate change response (Ebi *et al* 2021, Sheshadri *et al* 2021, Bouallègue *et al* 2024). Through this work, we propose a model-agnostic method providing calibrated error bounds for all variables, lead times, and spatial locations, requiring no model modifications with negligible computational costs.

Outline

The paper is structured as follows: section 2 introduces the inductive CP framework, spatio-temporal data, associated exchangeability assumptions, and the mathematical extension of CP to spatio-temporal domains. Section 3 presents experiments deploying our CP framework across diverse modelling tasks, from PDEs to climate modelling. Section 4 discusses the framework's strengths and limitations, and section 5 concludes.

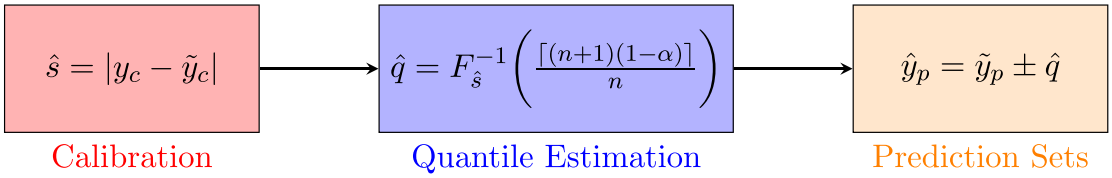


Figure 1. Inductive CP framework using absolute error residual (AER) nonconformity scores (see section 2.2): (1) Calibrate by computing nonconformity scores ($\hat{s}$) from calibration predictions ($\tilde{y}_c$) and targets (y_c). (2) Estimate the quantile ($\hat{q}$) for desired coverage $(1-\alpha)$ using n calibration samples and the inverse CDF $F_{\hat{s}}^{-1}$. (3) Construct prediction sets by applying $\hat{q}$ to test predictions ($\tilde{y}_p$).

2. CP

CP (Vovk *et al* 2005, Shafer and Vovk 2008) addresses a fundamental question in machine learning: given a dataset $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ and a trained model $\hat{f}: \mathcal{X} \to \mathcal{Y}$, how accurate is $\hat{f}$ at predicting the true label Y_{n+1} for a new query point X_{n+1}? CP extends point predictions $\hat{y}$ to prediction sets $\mathbb{C}^\alpha$ that contain the true label with guaranteed probability:

$$\mathbb{P}(Y_{n+1} \in \mathbb{C}^\alpha) \geqslant 1 - \alpha. \tag{1}$$

This coverage guarantee holds regardless of the model architecture or training procedure, requiring only that calibration samples are exchangeable (a weaker form of i.i.d)(Vovk *et al* 2005).

Several CP variants exist, with inductive CP (Papadopoulos 2008) being particularly efficient for neural networks. This approach splits data into a training set (for model training) and a calibration set (for constructing prediction sets $\mathbb{C}^\alpha$). The prediction sets satisfy equation (1) by comparing model outputs to calibration data using a *nonconformity score*, i.e. a metric quantifying prediction error.

The inductive CP framework operates through three fundamental steps (figure 1). First, *nonconformity scores* $s(x, y)$ are computed on calibration data, quantifying the disagreement between predictions and ground truth-larger scores indicate worse model performance. Second, the $(1-\alpha)$-quantile $\hat{q}$ of these scores is estimated, providing the threshold for constructing prediction sets. Third, for any test point x_{test}, the prediction set $\mathcal{C}(x_{\text{test}}) = \{y : s(x_{\text{test}}, y) \leqslant \hat{q}\}$ includes all outputs whose nonconformity scores fall below $\hat{q}$ (Angelopoulos and Bates 2023). Critically, while the coverage guarantee in equation (1) holds universally, regardless of model quality, the *usefulness* of the prediction sets depends entirely on the choice of nonconformity score function. Well-designed scores that accurately rank prediction difficulty yield tight intervals for easy inputs and wider intervals for challenging ones; poorly chosen scores produce uninformative but still valid prediction sets.

2.1. CP for spatio-temporal data

While originally developed for single-output predictions (Vovk *et al* 2005), CP has been extended to spatio-temporal domains (Stankeviciute *et al* 2021, Xu and Xie 2021, Sun 2022, Xu *et al* 2023, Ma *et al* 2024). We consider models predicting the evolution of spatio-temporal fields in physical systems as found in PDE modelling, weather forecasting, and fusion diagnostics. Each modelling task is formulated as an initial value problem where calibration and prediction sets consist of input–output pairs characterised by initial conditions and their solutions.

Our framework performs calibration independently for each cell of the spatio-temporal tensor (figure 2), providing marginal coverage guarantees at every spatial and temporal location. This cell-wise approach yields upper and lower bounds for each point without explicitly modelling spatial correlations, instead relying on the neural network to capture spatial dependencies during training. The discretised spatio-temporal grid must remain consistent between calibration and prediction sets. We formalise this approach to provide statistically valid, dimension-independent marginal coverage for high-dimensional outputs, as demonstrated across diverse applications, including multi-physics systems and operational weather models.

2.1.1. Mathematical formulation

Consider a model $\hat{f}$ mapping initial temporal sequences of spatial fields $X \in \mathbb{R}^{T_{\text{in}} \times N_x \times N_y \times N_{\text{var}}}$ to future sequences $\tilde{Y} = \hat{f}(X) \in \mathbb{R}^{T_{\text{out}} \times N_x \times N_y \times N_{\text{var}}}$, where $T_{\text{in}}, T_{\text{out}}$ denote input and output time steps, N_x, N_y are spatial dimensions, and N_{var} is the number of variables. The calibration procedure $\hat{q} = \hat{C}(\tilde{Y}, Y)$ uses model predictions $\tilde{Y}$ and ground truth Y to compute quantiles $\hat{q} \in \mathbb{R}^{T_{\text{out}} \times N_x \times N_y \times N_{\text{var}}}$ in a point-wise manner. These quantiles define lower (L) and upper (U) bounds forming the prediction set $\mathbb{C}$, with L and

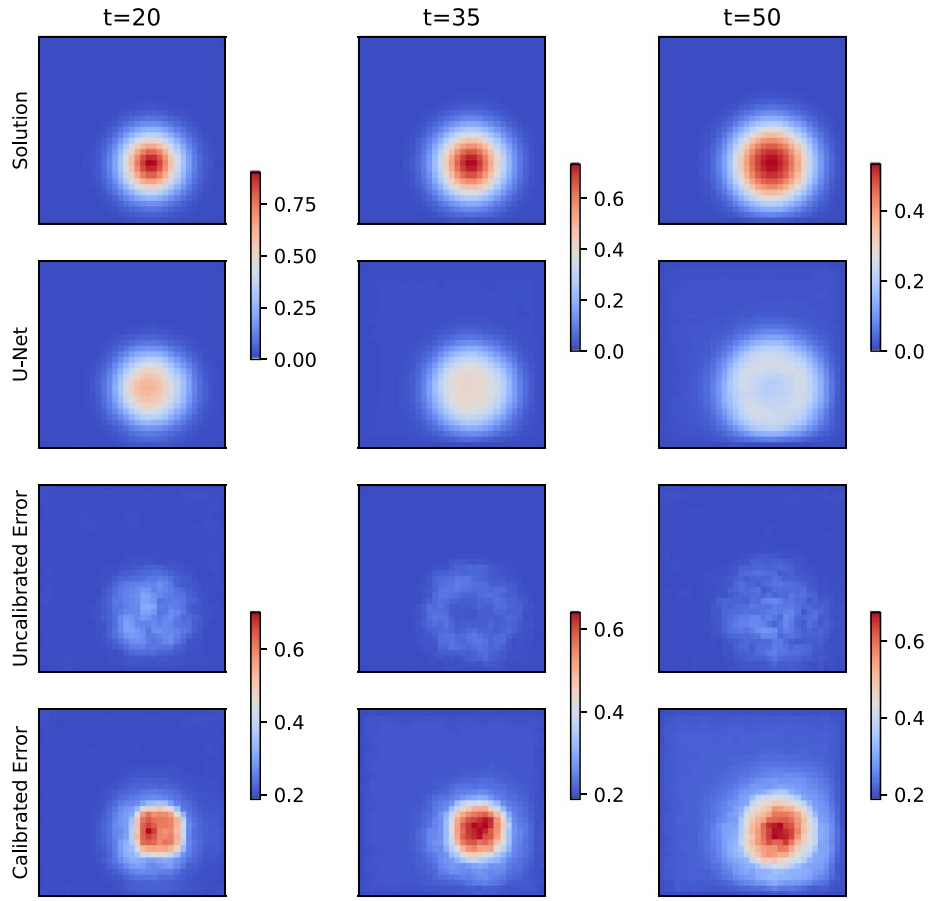


Figure 2. Cell-wise uncertainty calibration using CP for a U-Net modelling the wave equation in an out-of-distribution setting (section 3.3.1). Rows show: ground truth, model prediction, uncalibrated 95% coverage from MC dropout (2σ), and calibrated 95% coverage ($\alpha = 0.05$) from CP. Cell-wise calibration provides guaranteed coverage for each spatial location. MC dropout produces unrealistically small uncertainties, while CP error bars correctly identify regions of high uncertainty corresponding to complex dynamics.

U sharing the dimensionality of $\hat{q}$. For a test point X_{n+1} with true label Y_{n+1}, the coverage guarantee becomes:

$$\mathbb{E}\left[(Y_{n+1} \geqslant L) \wedge (Y_{n+1} \leqslant U)\right] \geqslant 1 - \alpha. \tag{2}$$

Equation (2) holds for each tensor cell given sufficient calibration samples and maintained exchangeability (Vovk 2012).

2.2. Nonconformity scores

Nonconformity scores quantify model deviation from ground truth using the calibration set (Angelopoulos and Bates 2023). We employ three methods:

- **Conformalised quantile regression (CQR)** (Romano *et al* 2019): Train three models to predict the $100 \times \alpha$th, median, and $100(1-\alpha)$th percentiles using quantile loss (Koenker 2005). The nonconformity score measures distance to the nearest bound: $s(x,y) = \max\{\underline{f}(x) - y, y - \bar{f}(x)\}$. After computing $\hat{q}$, the prediction set is obtained as $\{\underline{f}(x) - \hat{q}, \bar{f}(x) + \hat{q}\}$.
- **Absolute error residual (AER)** (Lei *et al* 2018): Train a single deterministic model using standard loss (e.g. MSE). Compute nonconformity scores as absolute errors: $s(x,y) = |y - \tilde{f}(x)|$. The prediction set becomes $\{\tilde{f}(x) - \hat{q}, \tilde{f}(x) + \hat{q}\}$. This requires no architectural modifications and is computationally efficient.
- **Standard deviation (STD)**: Use probabilistic models outputting mean $\mu(x)$ and STD $\sigma(x)$. The nonconformity score is $s(x,y) = \frac{|y-\mu(x)|}{\sigma(x)}$, yielding prediction sets $\{\mu(x) - \hat{q}\sigma(x), \mu(x) + \hat{q}\sigma(x)\}$. This requires architectural changes (e.g. dropout layers) or modified training (e.g. negative log-likelihood loss). The dependence on the STD of the prediction introduces a weak sense of conditionality.

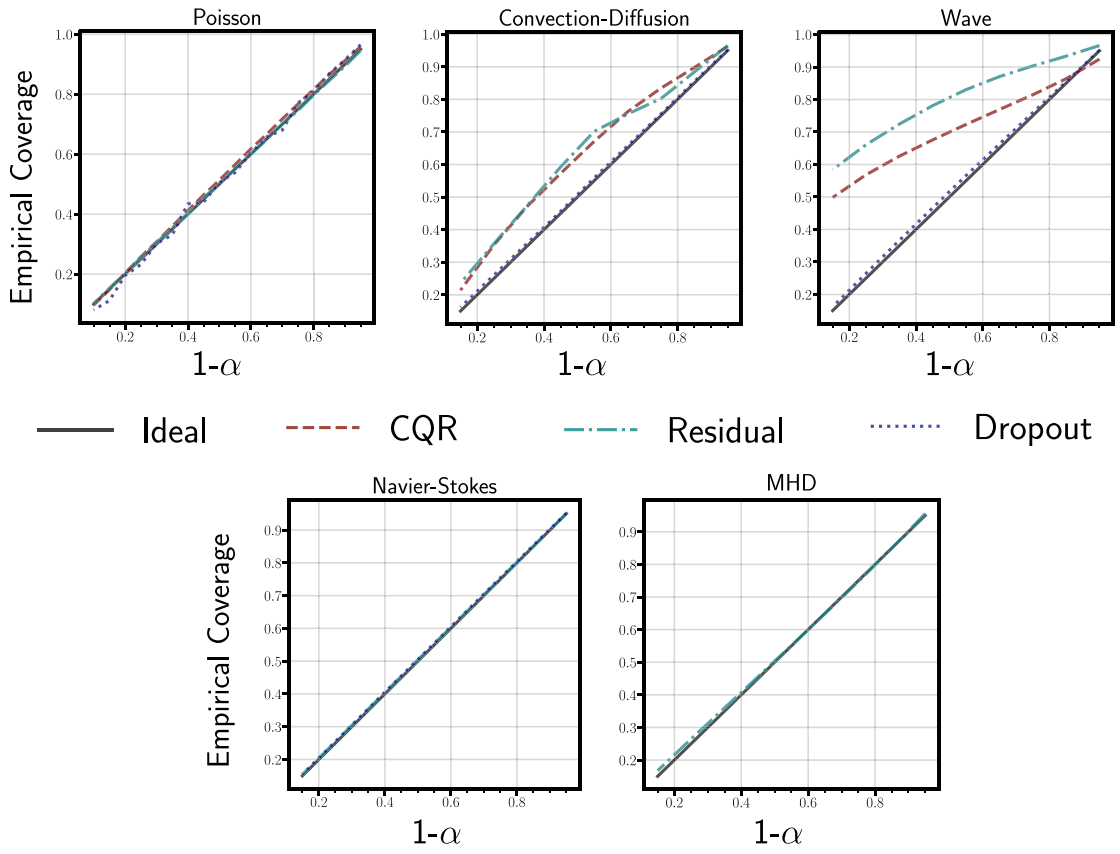


Figure 3. Empirical coverage versus target coverage $(1-\alpha)$ across experiments and nonconformity scores. The diagonal represents ideal coverage. All methods achieve near-perfect coverage across four PDE experiments, validating equation (1).

Figure 3 demonstrates that all three nonconformity scores achieve guaranteed coverage across diverse experiments (details in section 3). While coverage quality varies slightly between methods, CP ensures validity regardless of the choice. We select nonconformity scores based on practical considerations: architectural constraints, calibration cost, and data availability.

To validate coverage empirically, we compute:

$$\mathbb{P}\left(Y_{\text{val}} \in \mathbb{C}^{\alpha}\right) \approx \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} I_{\mathbb{C}^{\alpha}}\left(Y_i\right), \tag{3}$$

where $I_{\mathbb{C}^{\alpha}}$ is the indicator function for the prediction set. Valid coverage requires this to exceed $1-\alpha$. The empirical coverage obtained for a setting follows a Beta distribution characterised as (Vovk 2012):

$$\frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} I_{\mathbb{C}^{\alpha}}\left(Y_i\right) \sim \text{Beta}\left(n_{\text{cal}}+1-l, l\right), \tag{4}$$

where $l = \lfloor (n_{\text{cal}}+1)(1-\alpha) \rfloor$ and n_{cal} is the calibration set size.

2.3. Computational complexity of calibration

The CP calibration procedure has complexity $\mathcal{O}(d \cdot n_{\text{cal}} \log n_{\text{cal}})$, where $d = T_{\text{out}} \times N_x \times N_y \times N_{\text{var}}$ is the output dimensionality and n_{cal} is the calibration set size. This comprises: (1) computing nonconformity scores $s(x,y)$ for all n_{cal} samples across d dimensions, requiring $\mathcal{O}(n_{\text{cal}} \cdot d)$ element-wise operations (e.g. $|Y-\tilde{Y}|$ for AER), and (2) sorting scores to estimate quantiles $\hat{q}$ per dimension, requiring $\mathcal{O}(d \cdot n_{\text{cal}} \log n_{\text{cal}})$ operations. Once calibrated, constructing prediction sets for new predictions requires only $\mathcal{O}(d)$ operations by applying the pre-computed quantiles. The procedure requires only forward passes and sorting—no gradient computation or model retraining—and is embarrassingly parallel across dimensions. This computational efficiency is particularly advantageous compared to alternative UQ methods that require ensemble training ($\mathcal{O}(n_{\text{ensemble}} \cdot \text{training cost})$) or extensive Bayesian sampling. As shown in table 1, calibration remains practical even for $d > 10^7$ dimensions, with times ranging from

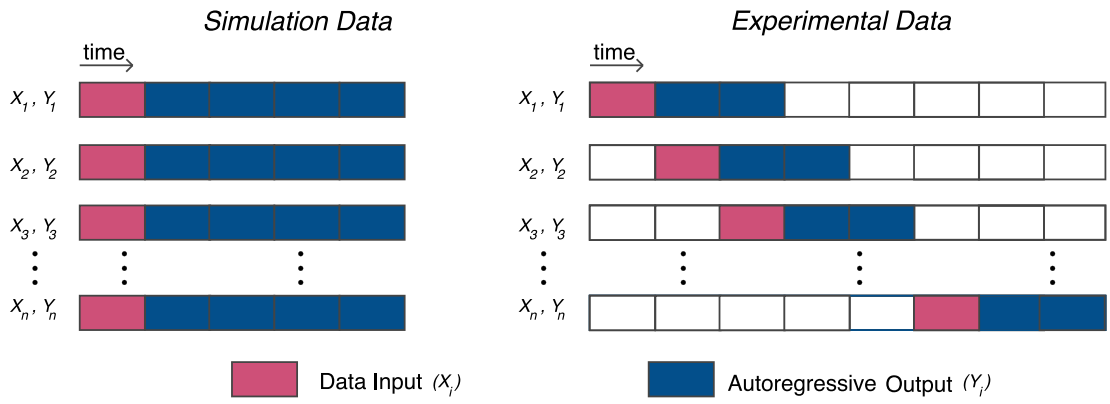


Figure 4. Constructing exchangeable pairs for simulation versus experimental data. *Simulation-based*: Multiple independent runs with varying initial conditions at $t = 0$, each forecasting the same time horizon to $t = T$. *Experimental data*: Single long-duration experiment partitioned into multiple IVPs with different starting times T_i from a larger temporal domain, each forecasting the same duration ΔT.

<1 second (low-dimensional cases) to a few hundred seconds (20 M+ dimensions for weather forecasting), demonstrating the near-zero computational cost that makes CP particularly suitable for production deployment of surrogate models.

2.4. Exchangeability requirements

We treat spatio-temporal surrogate modelling as an initial-value problem (IVP), where models evolve initial states autoregressively or in one-shot mappings. Each input–output pair (X_i, Y_i) from calibration and prediction sets is assumed exchangeable when initial conditions are sampled i.i.d. from the distribution of interest. This assumption requires: (1) consistent spatio-temporal structure (section 2.1.1) across the inputs and output, (2) identical discretised domains across calibration and prediction, and (3) exchangeability between calibration and test data (Angelopoulos and Bates 2023).

Our experiments (section 3) involve two distinct scenarios for constructing exchangeable pairs (figure 4):

2.4.1. Simulation-based exchangeability

For surrogate models of numerical simulations (sections 3.1–3.6), exchangeability is straightforward. The modelling task maps from the simulation start $t = 0$ to a fixed future time $t = T$. Each simulation begins at $t = 0$ with initial conditions sampled from a distribution $\mathcal{P}_{\text{IC}}$ characterising the parameter space of interest. For example, in the MHD plasma blob experiments (section 3.5), initial conditions vary in blob positions, widths, and amplitudes (table 9), while the temporal evolution window $[0, T]$ remains identical across all simulations.

This creates a natural exchangeability structure: each pair (X_i, Y_i) represents an independent draw from the joint distribution of initial conditions and their corresponding solutions. Multiple simulations with varied physical parameters provide abundant exchangeable pairs without violating temporal dependencies, as each simulation is an independent realisation of the physical system. The computational cost of generating additional calibration data is primarily limited by simulation expense rather than data availability.

2.4.2. Experimental data exchangeability

For surrogate models trained on experimental observations (sections 3.7 and 3.8), exchangeability is constructed through temporal windowing while preserving the IVP structure. Experimental data, such as continuous weather observations or plasma diagnostics, form long time series from which we extract multiple training examples.

The key insight is constructing exchangeable IVPs from a single temporal sequence. Each pair (X_i, Y_i) represents an IVP where: (1) initial condition X_i is extracted at time $t = T_i$ from the larger domain, (2) the model forecasts forward for fixed duration ΔT, (3) target Y_i spans $[T_i, T_i + \Delta T]$, and (4) starting

time T_i varies while ΔT remains constant. Here, these pairs can be treated as exchangeable, as we consider the data distribution to be characterised by a wide range of initial conditions, and the prediction window remains the same.

Crucially, predictions depend *only* on X_i, independent of absolute time T_i. A weather forecast initialised at 12:00 January 5th, predicting 48 hours ahead is exchangeable with one initialised at 18:00 January 12th with the same forecast horizon, such that both solve the same IVP: "given these atmospheric conditions, predict evolution over 48 hours." This time windowing transforms a single time series into N exchangeable pairs by sampling initial conditions $T_1, T_2, \ldots, T_N$ from distribution $\mathcal{P}_T$ representing typical states within the observed period. The fixed window ΔT ensures identical structure across pairs, satisfying section 2.1.1.

For fusion camera diagnostics (section 3.8), we extract 10-frame initial conditions from different time points within discharge shots, each predicting the subsequent 10 frames. The model learns mappings from the current plasma state to the near-future state, agnostic to absolute discharge time. Similarly, weather forecasting (section 3.7) constructs pairs from different initialisation times across months, each representing the same multi-day forecast problem.

Important limitation: Exchangeability can be violated with experimental data when calibration and test distributions differ (e.g. different seasons in weather, different plasma regimes in fusion). We demonstrate this sensitivity in sections 3.7 and 3.8, showing reduced coverage when exchangeability assumptions break. Users must verify distributional similarity or recalibrate when conditions change substantially.

3. Experiments

We empirically validate our CP framework across diverse surrogate models trained on spatio-temporal data from physical systems. Our experiments span multiple neural architectures commonly deployed in scientific modelling: Multi-layer perceptrons (MLPs) (Haykin 1994), U-Nets (Ronneberger *et al* 2015), Fourier neural operators (FNOs) (Li *et al* 2021), vision transformers (Yin *et al* 2022b), and graph neural networks (GNN) (Scarselli *et al* 2009). These architectures have proven effective for surrogate modelling in diverse applications, including wind turbine design (Lalonde *et al* 2021), high-energy physics (Baldi *et al* 2016), fusion reactors (Mánek *et al* 2023), fluid dynamics (Gupta and Brandstetter 2023), carbon capture (Wen *et al* 2023), weather forecasting (Kurth *et al* 2023, Lam *et al* 2023), and plasma evolution (Gopakumar *et al* 2023).

Relationship to Other UQ Methods: In this work, we focus on empirically demonstrating how inductive CP provides calibrated error bars across diverse spatio-temporal models using various non-conformity scores (as detailed in section 2.2). Our framework addresses both deterministic models (using AER) and probabilistic models-whether frequentist (CQR) or Bayesian (STD)-showcasing CP's model-agnostic nature and its ability to provide or calibrate guaranteed coverage across these different paradigms.

CP fundamentally differs from alternative UQ approaches such as deep ensembles and Bayesian neural networks by providing finite-sample, distribution-free guarantees on coverage. In contrast, ensemble and Bayesian methods provide asymptotic or model-dependent uncertainty estimates without such guarantees. While these methods can produce useful uncertainty estimates, they often fail to achieve desired coverage without post-hoc calibration. A comprehensive empirical comparison between CP and Bayesian deep learning methods for surrogate modelling-including coverage reliability, computational costs, and calibration quality-is presented in Gopakumar *et al* (2025), where we demonstrate that methods like MC Dropout and deep ensembles frequently require further calibration to achieve valid coverage. The present work establishes CP's applicability and scalability to high-dimensional spatio-temporal problems across multiple scientific domains.

Computational setup: All models were trained on NVIDIA A100 GPUs with 80GB memory. Calibration and prediction sets were evaluated on standard laptop hardware, demonstrating the computational efficiency of our approach. The experiment procedure is outlined in the algorithm below.

CP Framework: Experiment Structure CP framework structure.
1: Generate/gather training data (simulation or experimental)
2: Train surrogate model (or use pre-trained model)
3: Generate/gather calibration dataset (or use fine-tuning dataset)
4: Compute nonconformity scores and quantiles via CP framework
5: Construct prediction sets with guaranteed coverage
6: Validate coverage on independent test set

Table 1. Comprehensive coverage results for $\alpha = 0.1$ (90% target coverage). Uncalibrated coverage (unavailable for AER) shows initial estimates before CP calibration. Calibration time is reported on standard laptop hardware. Tightness represents average error bar width in normalised units (Min–Max normalisation between −1 and 1).

Case	Model	Output Dims	Method	Uncalib. (%)	Calib. (%)	Cal. Time (s)	Tightness
1D Poisson	MLP	32	CQR	94.61	90.01	0.0035	0.012
			AER	—	90.05	0.0030	0.002
			STD	97.5	90.85	0.133	0.025
1D Conv-Diff	U-Net	2000	CQR	25.53	93.05	19.70	0.314
			AER	—	92.60	8.30	0.266
			STD	88.43	90.29	88.15	0.164
2D Wave	U-Net	32 670	CQR	96.95	89.21	8.40	0.132
			AER	—	94.91	3.52	0.013
			STD	4.45	90.30	39.51	0.012
	FNO*	65 340	AER	—	89.24	34.18	0.330
			STD	32.81	89.83	462.0	0.669
2D Navier–Stokes	FNO*	40 960	AER	—	90.08	4.83	0.381
			STD	7.52	90.27	64.75	0.448
2D MHD	FNO	1348 320	AER	—	90.18	359.12	0.039
2D MHD	ViT* (PT)	313 344	AER	—	89.95	2980.50	0.062
	ViT* (FT)		AER	—	89.78	2078.50	0.015
2D Camera	FNO	2867 200	AER	—	91.28	293.62	0.131
2D Weather (Limited area)	GNN	20 602 232	AER	—	91.19	229.23	1.13
	GNN		STD	73.55	89.97	309.55	0.91
2D Weather (Global)	GNN	12 777 600	AER	—	90.03	366.41	1.34
	GNN		STD	71.22	89.88	400.57	1.28

*indicates out-of-distribution evaluation where calibration and training distributions differ. PT: pre-trained model; FT: fine-tuned model.

Table 1 presents our main empirical findings. Across all experiments, models, and nonconformity scores, we achieve near-perfect calibration to the target 90% coverage ($\alpha = 0.1$). For methods with initial uncertainty estimates (CQR and STD), we show both uncalibrated and calibrated coverage, demonstrating how CP corrects potentially misleading uncertainty quantification. The tightness metric, computed as average error bar width in the normalised space (linear transformation of the field to lie between −1 and 1), reveals that AER generally provides the tightest fits, with STD as a close second. Critically, these results hold across output dimensions ranging from 32 to over 20 million, demonstrating that our framework overcomes the curse of dimensionality. All models were trained on an Nvidia A100 GPU and evaluated and calibrated over standard laptop hardware.

3.1. 1D poisson equation

The Poisson equation generalises the Laplace equation and models diverse phenomena, including electrostatics, gravitation, and fluid potential fields (Hackbusch 2017). This steady-state elliptic PDE serves as our simplest test case, mapping an initial field distribution to its equilibrium state along a 1D domain $[0, 1]$ discretised into 32 uniform points.seasons in weather, differen

Dataset: We generated 7000 simulations using finite difference methods (py-pde package (Zwicker 2020)) by varying the initial field value $u_{\text{init}} \sim \mathcal{U}(0, 4)$, allocated as 5000 training, 1000 calibration, and

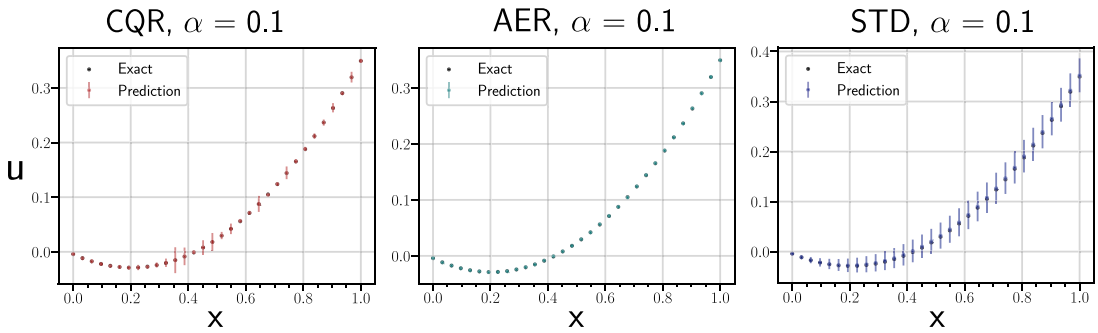


Figure 5. Calibrated prediction sets ($\alpha = 0.1$, 90% coverage) for the 1D Poisson equation using three nonconformity scores (CQR, AER, STD). The simple dynamics enable near-perfect model fit, resulting in tight, well-calibrated error bars. Ground truth (black line), model prediction (blue line), and shaded regions depict the prediction sets.

1000 validation samples. All datasets were sampled from the same distribution.seasons in weather, differen

Models and training: We trained separate MLPs (3 layers, 64 neurons per layer) for each nonconformity method: (i) three models for CQR modelling the 5th, 50th, and 95th quantiles using quantile loss (Koenker 2005); (ii) one deterministic model for AER using L1 loss; and (iii) one probabilistic model with dropout layers for STD. Training used the Adam optimiser (Kingma and Ba 2015) with initial learning rate 0.005 (halved every 100 epochs) for up to 1000 epochs. Further details about the physics, data generation strategies and model training can be found in appendix A. seasons in weather, differen

Results: Figure 5 visualises the $\alpha = 0.1$ prediction sets for all three methods. The MLP learns this simple mapping with high accuracy, yielding tight uncertainty bounds. All methods achieve the guaranteed coverage (table 1), with AER providing the tightest fit (0.002 normalised units) and minimal calibration time (3 ms). The probabilistic STD method achieves comparable coverage but with slightly wider bounds (0.025 normalised units) and longer calibration time (133 ms) due to Monte Carlo dropout sampling.

3.2. 1D convection–diffusion equation

We advance to a spatio-temporal system governed by the convection–diffusion equation, which combines parabolic and hyperbolic PDE characteristics to model transport phenomena across diverse applications (Chandrasekhar 1943). This equation describes how a fluid density field evolves under the competing effects of diffusion (smoothing) and convection (transport). We consider a spatially-varying diffusion coefficient and Gaussian initial conditions parametrised by mean and variance, with the system evolving over a 1D spatial domain $x \in [0, 10]$ and time interval $t \in [0, 0.1]$.seasons in weather, differen

Dataset: We generated 5000 simulations using a forward-time centred-space finite difference scheme across 200 spatial points and 100 time steps. The training set (3000 samples) was generated via Latin hypercube sampling over physically relevant ranges of diffusion coefficients, convection velocities, and initial condition parameters. To test out-of-distribution robustness, calibration and validation sets (1000 each) were sampled from a shifted parameter regime with reduced diffusion and enhanced convection. This distribution shift mimics real-world deployment scenarios where test conditions may differ from training.seasons in weather, differen

Model and training: A 1D U-Net with 4 encoder–decoder levels maps the first 10 time steps (downsampled from 100) to the next 10 steps. The architecture uses batch normalisation and tanh activations. For STD, we added dropout layers (rate 0.1) after each encoder–decoder block. Training followed the same optimiser schedule as the Poisson case, using quantile loss for CQR and MSE loss otherwise. Further details about the physics, data generation strategies and model training can be found in appendix B. seasons in weather, differen

Results: Despite testing on a physical regime with 2× higher convection and half the diffusion compared to training data (figure 6), CP provides valid guaranteed coverage (table 1). The uncalibrated CQR severely underestimates uncertainty (25.53% coverage), while CP calibration increases this to 93.05%. For STD, the uncalibrated 88.43% coverage is refined to 90.29%. Figure 3 confirms guaranteed coverage across all α levels for both training-distribution and out-of-distribution settings. While CQR and AER produce conservative bounds (potentially due to model over-fitting on this relatively simple

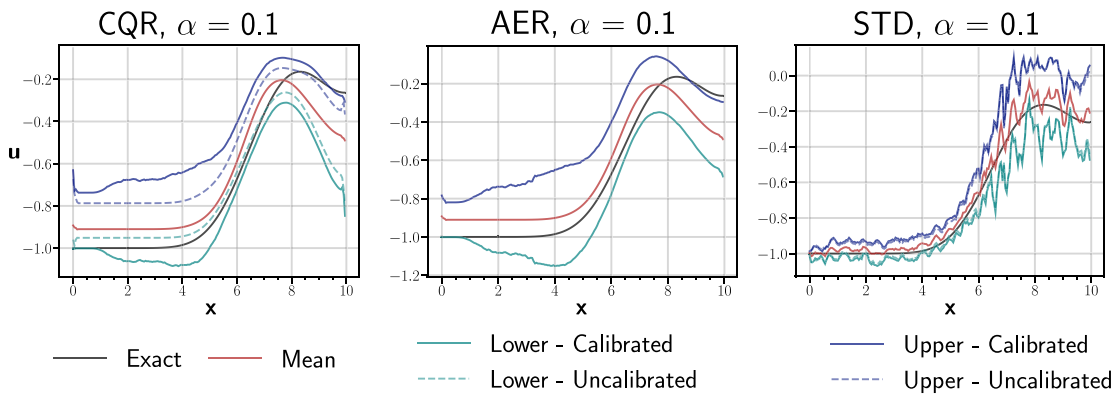


Figure 6. Calibrated prediction sets ($\alpha = 0.1$, 90% coverage) for 1D convection–diffusion evaluated on out-of-distribution data at final time $t = 0.1$. CP successfully calibrates initially insufficient coverage (CQR: 25.53% → 93.05%) or refines sampled uncertainty (STD: 88.43% → 90.29%) to guarantee target coverage. Ground truth (black), prediction (blue), calibrated bounds (red shaded), and uncalibrated bounds (gray dashed) where applicable.

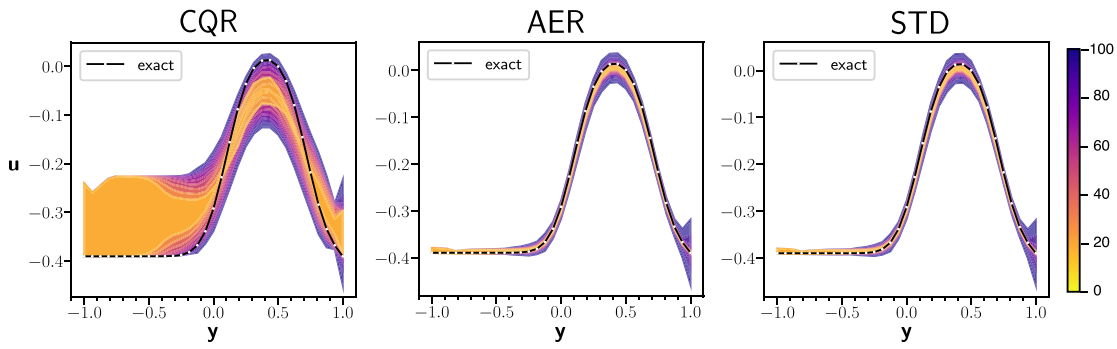


Figure 7. Prediction sets for the 2D wave equation using three nonconformity scores. Colour bars show coverage levels (in %, corresponding to $1 - \alpha$). Each panel displays a spatial slice along the y-axis at the final time step. The ground truth (black with white markers) lies within the calibrated bounds, validating equation (1).

task), all methods maintain coverage guarantees. This experiment demonstrates a critical capability: **CP provides valid uncertainty quantification even when deployed outside the training distribution**, provided exchangeability holds between calibration and prediction regimes.

3.3. 2D wave equation

The 2D wave equation models wave propagation in acoustics, optics, and quantum mechanics (Tipler 2008). We simulate Gaussian wave packets evolving on a 33×33 spatial grid over 80 time steps. A training dataset of 500 simulations is generated by varying the amplitude and position of the initial Gaussian. We train both a U-Net (feed-forward) and an FNO (autoregressive) to model the temporal evolution. Calibration and validation each use 100 additional simulations respectively. Full physics details and numerical methods are in appendix C.

3.3.1. U-net

The U-Net performs a single feed-forward mapping from 20 input time steps to 30 output time steps, producing outputs of shape $[30, 33, 33]$. Coverage is estimated cell-wise across the output tensor following section 2.1.1.

Figure 7 shows spatial slices of prediction sets at multiple α levels. All three methods achieve valid coverage (figure 3 and table 1), with AER being computationally cheapest. Both CQR and AER produce conservative (wide) intervals, as expected from the inequality in equation (1). AER and STD provide tighter fits than CQR while maintaining coverage guarantees.

Out-of-distribution testing. To test robustness, we generate new calibration and validation datasets by solving the wave equation with half the wave velocity used during training. This tests CP's effectiveness when the calibration regime differs from the training distribution. As shown in figure 2, uncalibrated MC dropout fails to capture modelling errors in this out-of-distribution regime. In contrast, CP

provides statistically guaranteed bounds regardless of the model's training conditions. This demonstrates CP's value for deploying pre-trained surrogates in new physical regimes without costly retraining.

3.3.2. FNO

We train an FNO in an autoregressive framework: the model takes 20 initial time steps, predicts the next 10 steps, then recursively unrolls to produce 60 total output steps (shape $[60, 33, 33]$). CP is performed over the entire rolled-out output. Since FNOs perform best with relative L^p loss, we omit CQR for this architecture.

The FNO is tested in both in-distribution and out-of-distribution (half-speed) settings, matching the U-Net experiments. Figure 8 shows that CP provides valid error bars under both conditions. The key requirement is exchangeability between calibration and prediction regimes, not similarity to training data. This enables valuable UQ even for previously unseen solution families. The FNO achieves tighter coverage than the U-Net (figure 3), likely due to its operator learning formulation. Additional coverage plots are in figure 23(appendix C).

3.4. 2D Navier–Stokes equations

The incompressible 2D Navier–Stokes equations describe viscous fluid dynamics, modelling conservation of mass and momentum. Their complexity and strong nonlinearity necessitate CFD solvers. Neural-PDE methods offer efficient alternatives at scale (Azizzadenesheli *et al* 2024). Following Li *et al* (2021), we train an FNO to model vorticity evolution. The model is trained on simulations with viscosity $\nu = 10^{-3}$, then calibrated and tested on data with $\nu = 10^{-4}$ (out-of-distribution). The FNO maps 10 input time steps to the next 10 output steps. Physics details and training specifications are in appendix D.

To enable STD-based CP, we modify the FNO architecture by adding dropout layers, creating a probabilistic operator. The resulting model outputs both mean predictions and uncertainty estimates via MC dropout sampling.

Figure 9 demonstrates CP's ability to calibrate probabilistic models. The uncalibrated dropout-based uncertainty provides only 7.52% coverage, severely underestimating prediction errors. CP adjusts these intervals to achieve the target 67% coverage (we show 67% instead of the standard 90% to better visualise the calibration effect at moderate coverage levels). Coverage validation across all α levels is shown in figure 3. This helps illustrate CP's dual utility: providing guarantees for deterministic models (via AER) and calibrating uncalibrated probabilistic models (via STD).

3.5. 2D magnetohydrodynamics (MHD)

MHD couples the Navier–Stokes equations with Maxwell's equations to model plasma evolution in fusion devices such as tokamaks (Bellan 2006). We consider a reduced-MHD system (Hoelzl *et al* 2021) describing multiple plasma blobs in a non-uniform temperature field. The system evolves three coupled variables—density (ρ), electrostatic potential (Φ), and temperature (T)—on a 106×106 toroidal grid (coordinates R, Z). This represents a challenging multi-variable, multi-physics problem. Full physics equations are in appendix E.

We use 2000 simulations from the JOREK code (Hoelzl *et al* 2021), split into 1000 for training, 500 for calibration, and 500 for validation. Each simulation varies the initial conditions (blob positions, widths, amplitudes) while keeping the physics parameters fixed. The dataset and pre-trained model are taken from Gopakumar *et al* (2024).

A multi-variable FNO learns the coupled dynamics of all three fields simultaneously. The model autoregressively predicts from 10 input time steps to 5 output steps, recursively continuing to the 50th time step. CP is performed over the complete spatio-temporal domain, yielding prediction sets for each cell in the 4D output tensor (time $\times$ R $\times$ Z $\times$ variables). Given the model's scale (9.4 M parameters, 1.35 M output dimensions), we use only the AER method for computational efficiency.

Figure 10 shows spatial slices through the prediction sets for all three variables at a single time step. The calibrated error bars successfully bound the sharp density, potential, and temperature peaks characterising the plasma blobs. Coverage validation (figure 3) confirms guaranteed marginal coverage across all 1348 320 output dimensions ($50 \times 106 \times 106 \times 3$), demonstrating CP's scalability and immunity to the curse of dimensionality. The bounds provide interpretable confidence estimates for each variable's spatial distribution, enabling assessment of the surrogate's reliability across the operational domain.

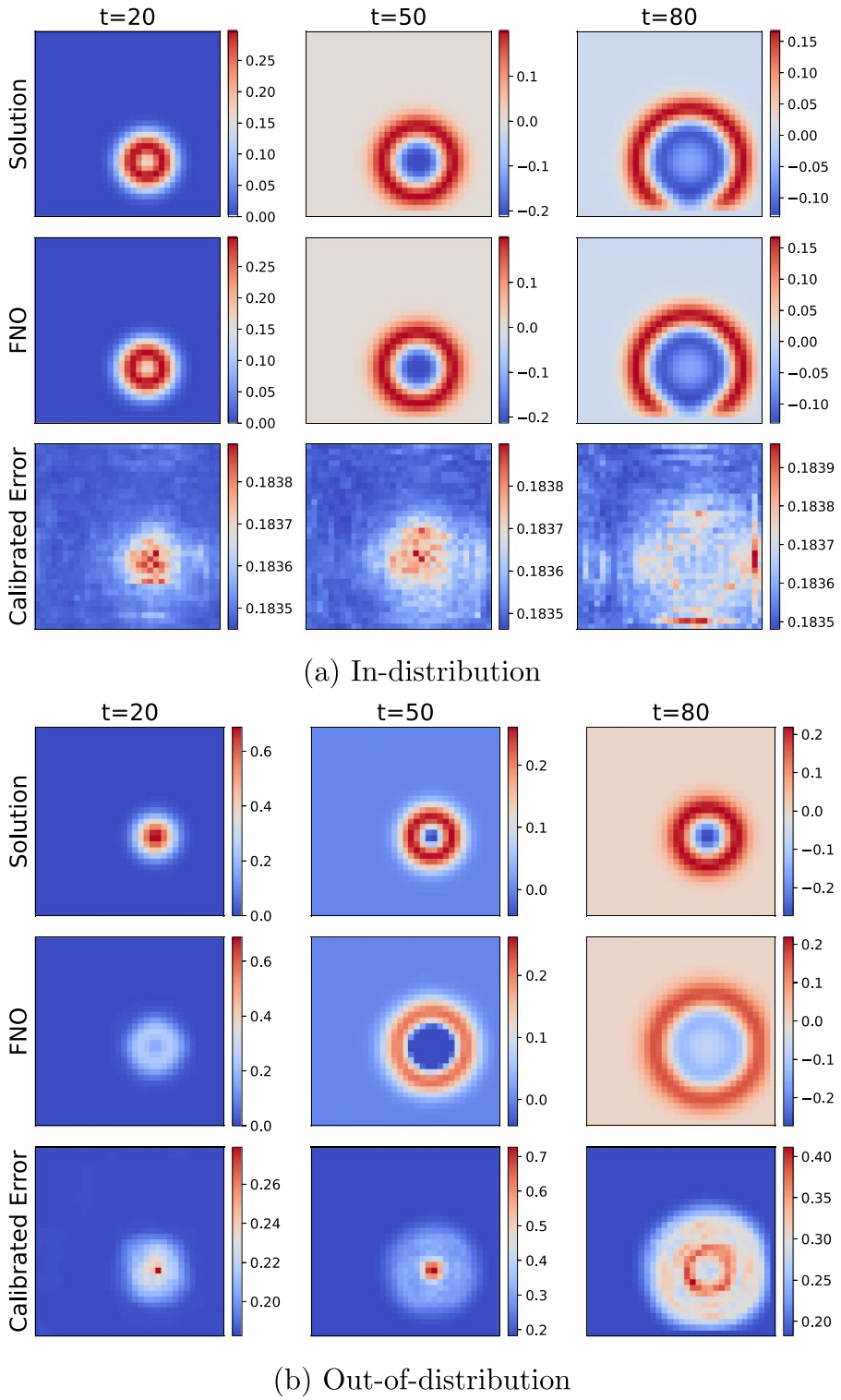


Figure 8. CP performance for autoregressive FNO on the wave equation. Top row: ground truth; middle row: FNO predictions; bottom row: calibrated error bars (90% coverage). Valid coverage is maintained both within (a) and outside (b) the training distribution, demonstrating CP's robustness to distribution shift.

3.6. Foundational physics models

Foundation models pre-trained on diverse PDE datasets (Bommasani *et al* 2022, McCabe *et al* 2023, Alkin *et al* 2024, Hao *et al* 2024, Rahman *et al* 2024) have emerged as a promising approach for multi-task scientific modelling. These models employ transformer-based architectures with attention mechanisms across spatio-temporal domains, learning shared representations of differential operators (e.g. diffusion, convection) common across PDE families. This enables them to capture global behaviours during pre-training, leaving task-specific local features for fine-tuning (Alkin *et al* 2024).

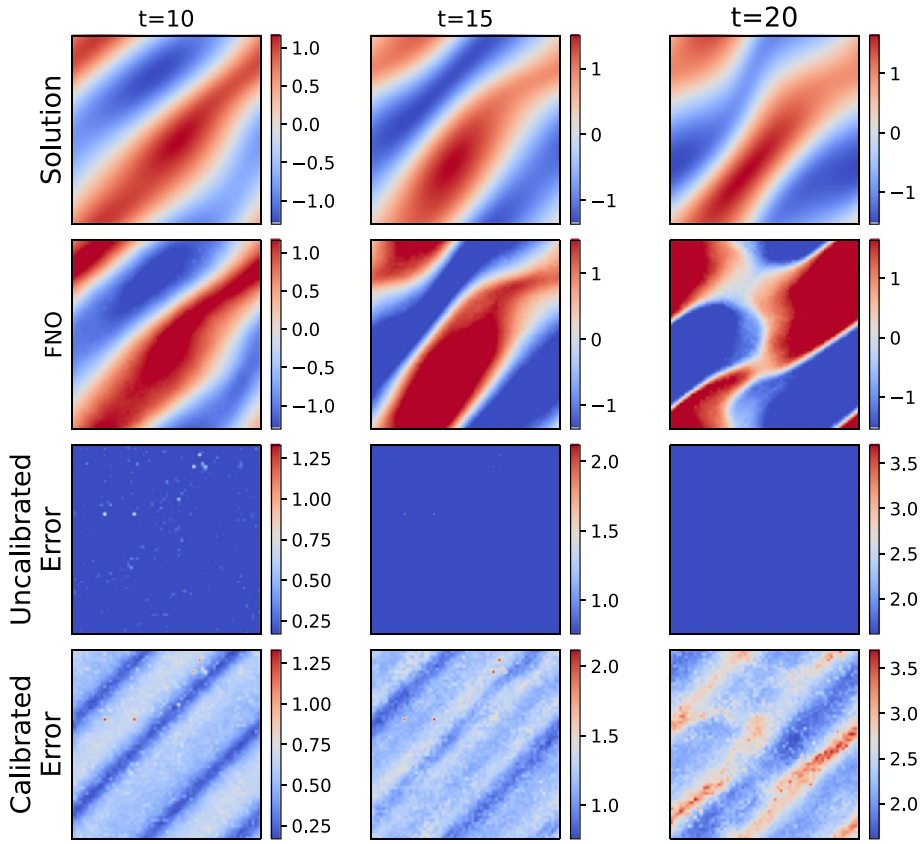


Figure 9. Calibrating probabilistic FNO predictions for out-of-distribution Navier–Stokes data. Rows show: ground truth, FNO predictions, uncalibrated MC dropout uncertainty (standard deviation), and calibrated CP error bars (67% coverage). The uncalibrated dropout underestimates errors, but CP corrects this to achieve the target coverage.

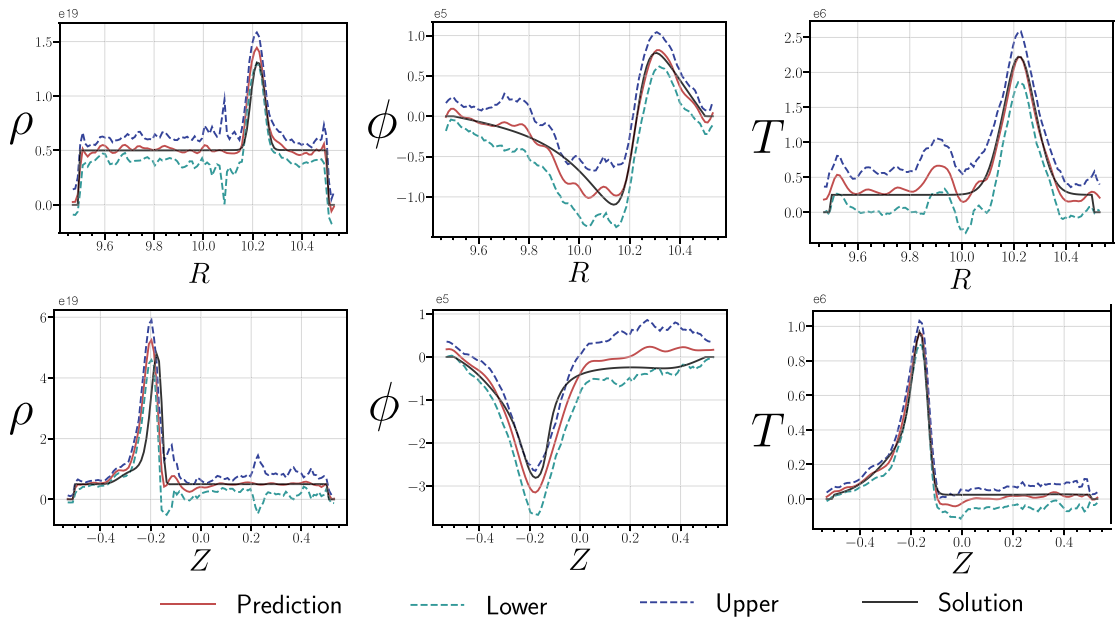


Figure 10. Spatial profiles at timestep 20 showing 90% coverage ($\alpha = 0.1$) for multi-variable MHD predictions. Columns: density ρ (left), potential Φ (centre), temperature T (right). Rows show profiles along the R-axis (top) and Z-axis (bottom). For each variable, we display the numerical solution, FNO prediction, and calibrated upper/lower bounds. CP accurately captures the sharp features of plasma blobs across all three coupled variables.

As these models scale and deploy across safety-critical applications, UQ becomes essential. CP offers an efficient validation framework: for fine-tuned models, the existing fine-tuning data can serve as calibration data, eliminating the need for additional simulations. This is justified because fine-tuning aims

to align the model with a specific target distribution, making performance within that distribution the primary concern.

We apply CP to the multi-physics pre-trained adaptive vision transformer (MPP-AViT) of McCabe *et al* (2023). This model uses shared embeddings and normalisation across variables, with an AViT backbone (Yin *et al* 2022a) that sequentially attends over space and time. The model autoregressively predicts the next time step given current field values. We refer to McCabe *et al* (2023) for full architectural and training details.

3.6.1. Pre-trained model: zero-shot learning

We test the largest pre-trained model (MPP-AViT-L, 409 M parameters) on MHD density evolution—a physics regime not seen during training. The model was pre-trained on shallow-water, diffusion-reaction, and Navier–Stokes equations, learning to model densities, velocities, and pressures. We extract only density fields from our MHD dataset (section 3.5) for inference. The model takes 16 input time steps and autoregressively predicts a single step forward until the 50th time step.

Figure 11 shows that despite zero exposure to MHD during training, the model captures the major features of plasma blob evolution (radial outward motion). However, finer details are lost, and patch-based artefacts appear, a known limitation of the architecture (McCabe *et al* 2023). Using 1000 calibration data points and the AER method, CP provides valid 95% coverage bounds (appendix F). Importantly, for this deterministic model, the calibrated errors ($\hat{q}$) are input-independent constants determined solely by the calibration data, representing global rather than instance-specific uncertainty.

3.6.2. Fine-tuned model

We fine-tune the smaller MPP-AViT-Ti variant on MHD density fields using 75% of the data for training/calibration and 25% for validation. Here, the training data serves a dual purpose as calibration data, justified because fine-tuning targets a specific distribution, and we only care about performance within that regime.

Figure 12 demonstrates substantial improvement over zero-shot performance. The fine-tuned model accurately captures both major and fine-scale density features. CP provides guaranteed coverage (appendix F), though the error bars reflect global rather than local uncertainties due to the model's deterministic nature. For well-fitted deterministic models like this, CP identifies regions of generally high dynamics across the dataset rather than instance-specific failure modes. Probabilistic models using the STD method would provide input-dependent bounds that adapt to each specific prediction.

3.7. Neural weather prediction

In addition to surrogate models of systems described by explicit PDEs, the proposed methodology is also applicable to more general machine learning models describing physical processes. To demonstrate this, we here study the use of CP for data-driven weather forecasting models. Traditional weather forecasting models typically combine PDEs describing large-scale interactions and parametrisations describing subgrid-scale physical processes (Kalnay 2002). Data-driven machine learning models approximate this whole process with a single neural network model. This allows for orders of magnitude faster forecasting speed and, when training on data incorporating observations, also more accurate forecasts (Bi *et al* 2023, Kurth *et al* 2023, Lam *et al* 2023, Bouallègue *et al* 2024).

Due to the chaotic nature of the weather system, capturing uncertainty in weather forecasts has long been an important consideration both in research and operations. Such probabilistic modelling has typically been achieved by ensemble forecasting, where perturbations are used to produce samples of possible forecast trajectories (Coiffier 2011). Existing data-driven models are still largely deterministic (Rasp *et al* 2024). There are attempts to produce ensemble forecasts using machine learning models by perturbing initial states (Kurth *et al* 2023, Chen *et al* 2023b), training multiple models (Graubner *et al* 2022) or generative modelling (Hu *et al* 2023, Price *et al* 2023, Oskarsson *et al* 2024). Fundamentally, ensemble forecasting always requires a computational cost proportional to the number of ensemble members, i.e. the number of forecasts made via perturbations. In contrast, CP offers a cheap method to immediately quantify forecast uncertainty for a time, position, and variable of interest. This uncertainty can be used by meteorologists interpreting the forecast, conveyed to decision-makers reacting to extreme weather events or directly presented to end-users looking up the forecast for the coming week. As CP enables UQ for a single forecast output by the model, it is directly applicable to existing deterministic machine learning models. A limitation of scalar uncertainty estimates is that there are no samples of the distribution over the atmospheric state. In some scenarios, it can be valuable to inspect such samples to gain an understanding of how different weather scenarios are unfolding.

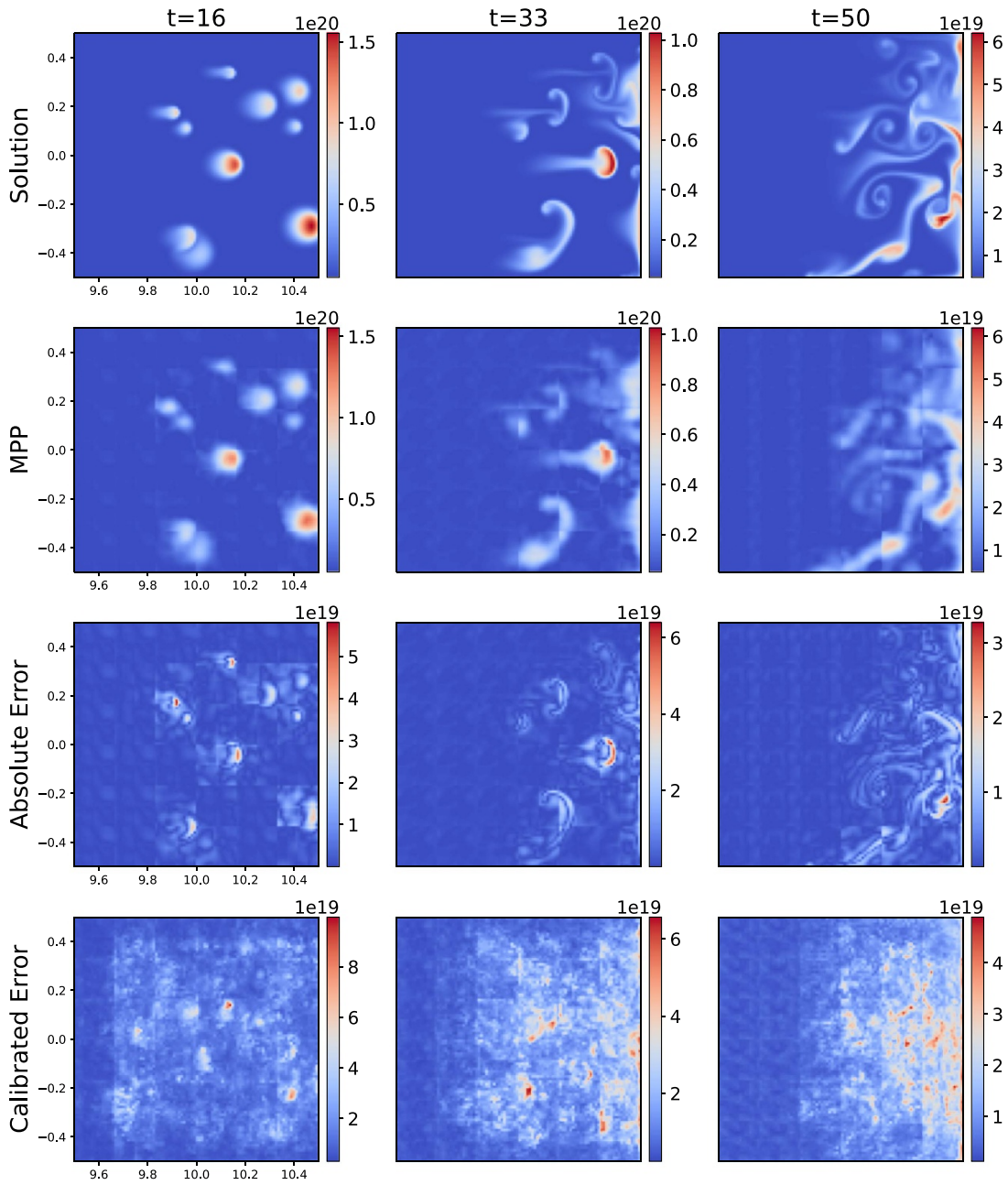


Figure 11. Zero-shot MPP-AViT performance on out-of-distribution MHD density evolution. Rows show: ground truth, model predictions, absolute error, and calibrated error bar width (95% coverage). Despite no training on MHD physics, the model captures major blob features. CP quantifies the prediction uncertainty with guaranteed coverage.

3.7.1. Model

We apply CP to the graph-FM model of Oskarsson *et al* (2024). Graph-FM is a graph-based neural weather prediction model (Keisler 2022, Lam *et al* 2023), where a hierarchical GNN is utilised for producing the forecast. Let X^t denote the full weather state at time step t, including multiple atmospheric variables modelled for all grid cells in some discretisation. Examples of such atmospheric variables are temperature, wind, geopotential and solar radiation. The GNN g in Graph-FM represents the single time step prediction

$$X^{t+1} = g\left(X^{t-1:t}, F^{t+1}\right) \tag{5}$$

where F^{t+1} are known forcing inputs that should not be predicted. Taking the two past states as inputs to g allows the model to make use of both magnitude and first derivative information. Equation (5) can be applied iteratively to roll out a complete forecast of T time steps. The full forecasting model can thus be viewed as a mapping from initial weather states $X^{-1:0}$ and forcing $F^{1:T}$ to a forecast $X^{1:T}$. The forecast

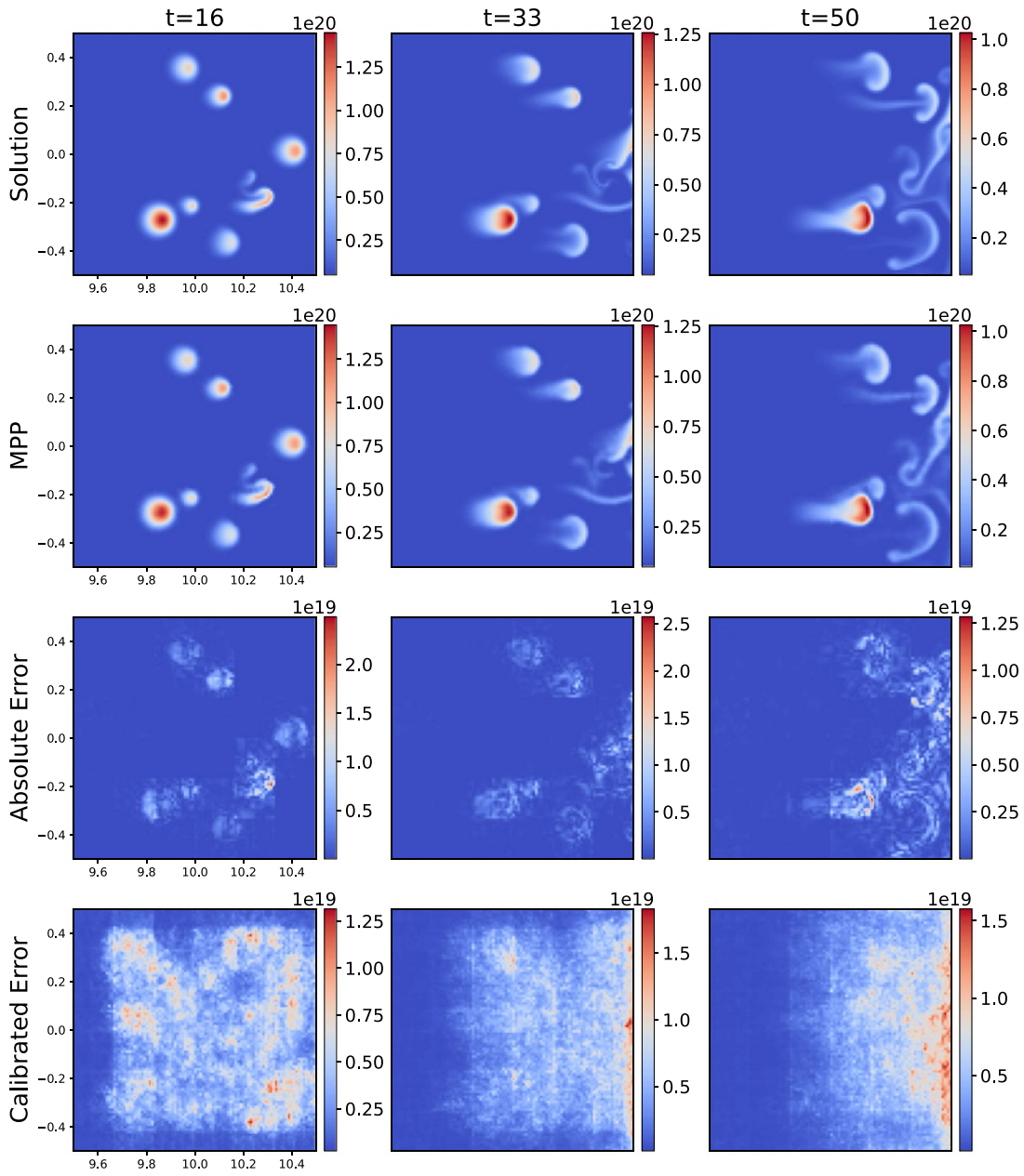


Figure 12. Fine-tuned MPP-AViT performance on MHD density. Rows show: ground truth, predictions, absolute error, and calibrated error bar width (95% coverage). Fine-tuning dramatically improves accuracy compared to zero-shot (figure 11). For this well-fitting deterministic model, calibrated errors capture global uncertainties in regions of high dynamics.

$X^{1:T}$ is a tensor of shape $T \times N_x \times N_y \times N_{\text{var.}}$ where $N_{\text{var.}}$ is the number of atmospheric variables modelled. We consider two versions of Graph-FM, trained with different loss functions:

- **Graph-FM (MSE):** Graph-FM trained with a weighted MSE loss. This model outputs only a single prediction, to be interpreted as the mean of the weather state.
- **Graph-FM negative log-likelihood (NLL):** A version of Graph-FM that outputs both the mean and STD for each time, variable and grid cell. This model was trained with a NLL loss, assuming a diagonal Gaussian predictive distribution (also referred to as the uncertainty loss (Chen *et al* 2023a)). Apart from the change of loss function, the training setup was identical.

For the Graph-FM (MSE) we compute non-conformity scores using the AER strategy. As the Graph-FM (NLL) is probabilistic, we use STD non-conformity scores. Note that these are computed based on the STDs directly output from the model, rather than from sample estimates based on MC dropout.

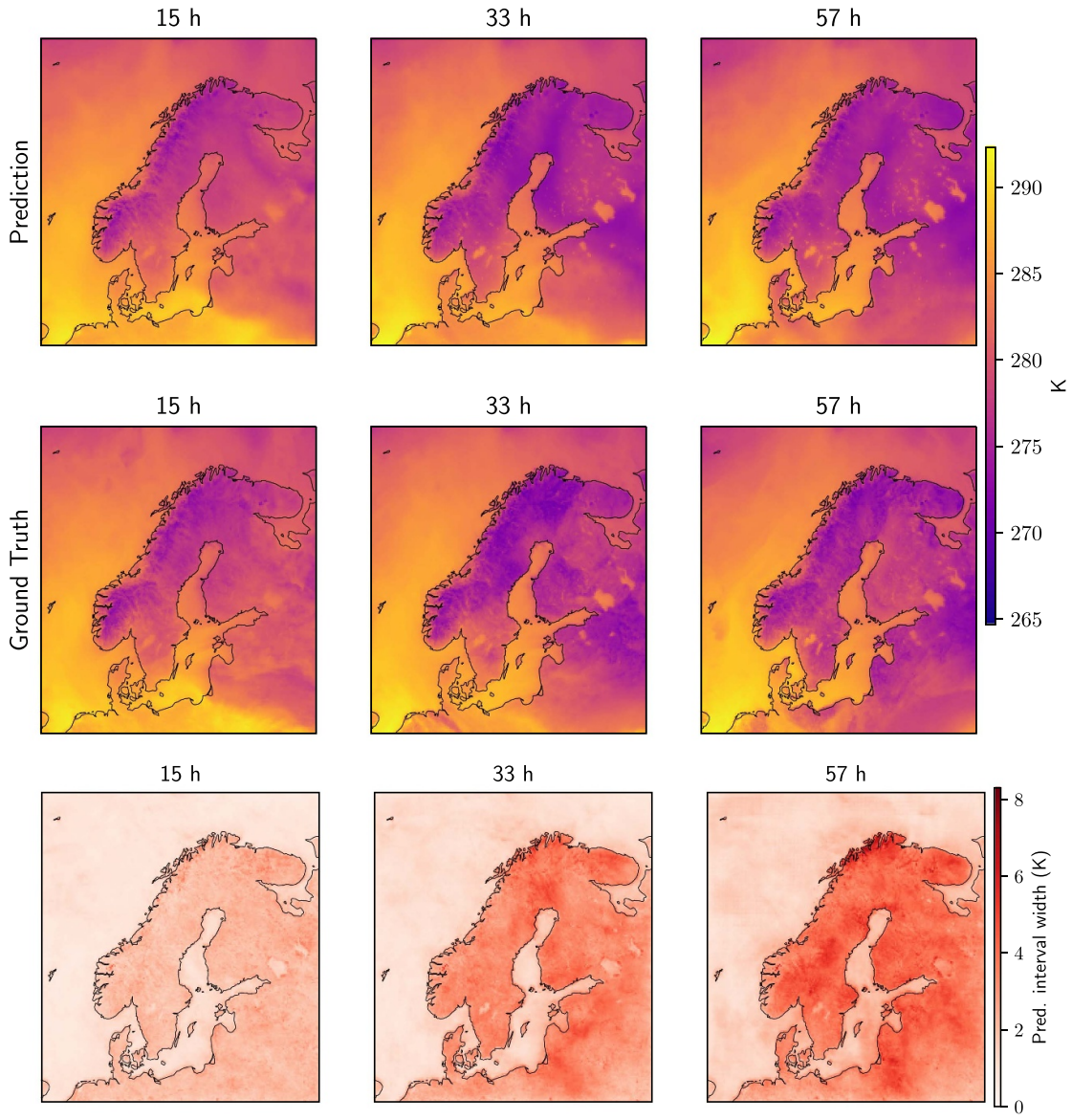


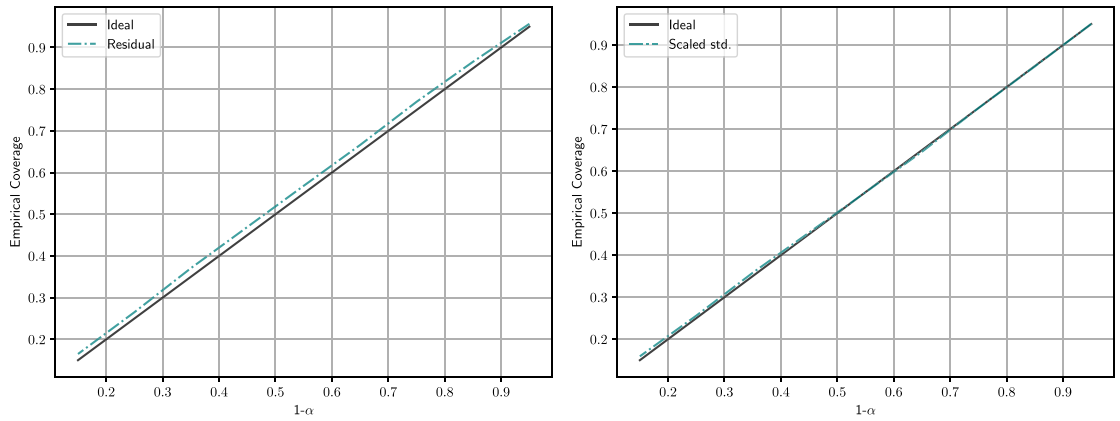
Figure 13. Prediction (top), ground truth (middle) and width of the error bars (bottom) at $\alpha = 0.05$ for predicting the temperature 2 m above ground (`2t`) using Graph-FM (MSE).

3.7.2. Limited area forecasting

In this first experiment, we apply CP to a limited area version of Graph-FM. Forecasts are here produced for a limited area covering the Nordic region. These Graph-FM models were trained on the limited area dataset from Oskarsson *et al* (2024), consisting of forecasts from the MEPS system (Müller *et al* 2017). One such forecast includes $N_{\text{var.}} = 17$ variables modelled on a $N_x \times N_y = 238 \times 268$ grid over $T = 19 \times 3$ h time steps (up to 57 hour lead time). When Graph-FM is used in a limited area configuration, it produces weather forecasts for a specific sub-area of the globe. To achieve this, boundary conditions along the edges of the forecasting area are given as important forcing inputs. The exact models used have 4 graph processing layers and use 64-dimensional latent representations. We refer to Oskarsson *et al* (2024) for further details about the model and data.

We use forecasts started during September 2021[6] As our calibration data set forecasts that started during September 2022 as test data. By using the same month for calibration and testing, we minimise the effect of distributional shifts due to seasonal effects. Having access to calibration data from the same month, collected the previous year, is a reasonable assumption in practical settings.

[6] For calibration, we specifically use forecasts started during the dates 04-09-2021–30-09-2021. The model was trained on forecasts started during the last days of August, which are rolled out over the first days of September. To avoid strong correlations to the training data, we use only forecasts from September 4 onwards for calibration.



(a) Graph-FM (MSE) (b) Graph-FM (NLL)

Figure 14. Empirical coverage for weather forecasting models.

Figure 13 shows the ground truth, predicted forecast and the conformalised error intervals for temperature 2 m above ground. Considering the autoregressive nature of Graph-FM, the error accumulates and grows further in time, which is accurately captured by the CP framework (refer figure 16).

Figure 14 shows the empirical coverage for the test set. With CP, we can achieve calibrated uncertainty estimates for both versions of Graph-FM. Of great interest in the weather forecasting setting is the uncertainty for specific future time points. We visualise this by plotting the width of the error bars for all spatial locations at different lead times in an example forecast. Such plots for shortwave solar radiation are shown in figure 15 and for geopotential in figure 16.

Figure 15 highlights an important difference between the two methods for computing non-conformity scores. As the shortwave solar radiation is close to 0 during the night, it is easy for the model to predict. During the day, this is far more challenging. With the AER non-conformity scores, used for Graph-FM (MSE) in figure 15(a), the width of the predictive intervals is determined during calibration, and does not change depending on the forecast from the model. As a specific lead time can fall both during the day and night, depending on the initialisation time, CP will give large error bars also during the night. This can be compared to the results for Graph-FM (NLL) in figure 15(b), using STD non-conformity scores. In this case, the bounds are very tight for lead times during the night (33 h and 57 h). It can also be noted that for Graph-FM (NLL) at lead time 15, we see clear spatial features appearing in the error bars themselves. This corresponds to higher forecast uncertainty in areas of rapid change. The conditional dependency that emerges while using STD in Graph-FM (NLL) thus has desirable properties, but this relies on having a model that outputs (potentially uncalibrated) STDs.

3.7.3. Global forecasting

We next experiment with CP for global weather forecasting. The models used are again Graph-FM (MSE) and Graph-FM (NLL), but here applied on the full globe. These models are trained on a version of the ERA5 reanalysis dataset (Hersbach *et al* 2020) using a 1.5 latitude-longitude grid. The global models have 8 graph processing layers and use 256-dimensional latent representations. Each forecast includes 5 surface variables and 6 atmospheric variables, each modelled at 13 different vertical pressure levels in the atmosphere. Due to the large number of variables forecast (83 in total), we here only perform CP for a subset of these. This subset includes all surface variables and the atmospheric variables at pressure level 700 hPa. This results in a total of $N_{\text{var.}} = 11$ variables, modelled on a $N_x \times N_y = 240 \times 121$ grid over $T = 40$ time steps (up to 10 days lead time with 6 h time steps). We note that a strength of the CP framework is that uncertainty quantification can be performed per variable, alleviating memory issues during calibration. Therefore, the procedure could trivially be extended to the full set of variables, as long as the full forecasts are stored. We again refer to Oskarsson *et al* (2024) for more details about the global models and data configuration.

For the global experiment, we use full years of forecasts for calibration and evaluation, all starting from ERA5 as initial conditions. Forecasts at 00 and 12 UTC each day of 2018 are used as the calibration set, and a similar set of forecasts for 2019 is used as the test set[7]. The ground truth is given

[7] We remove forecasts started during the last 10 days of 2018 from the calibration set to avoid strong correlations with the test data at the start of 2019. Note that forecasts in the test set started during the last days of 2019 will extend into time points in 2020.

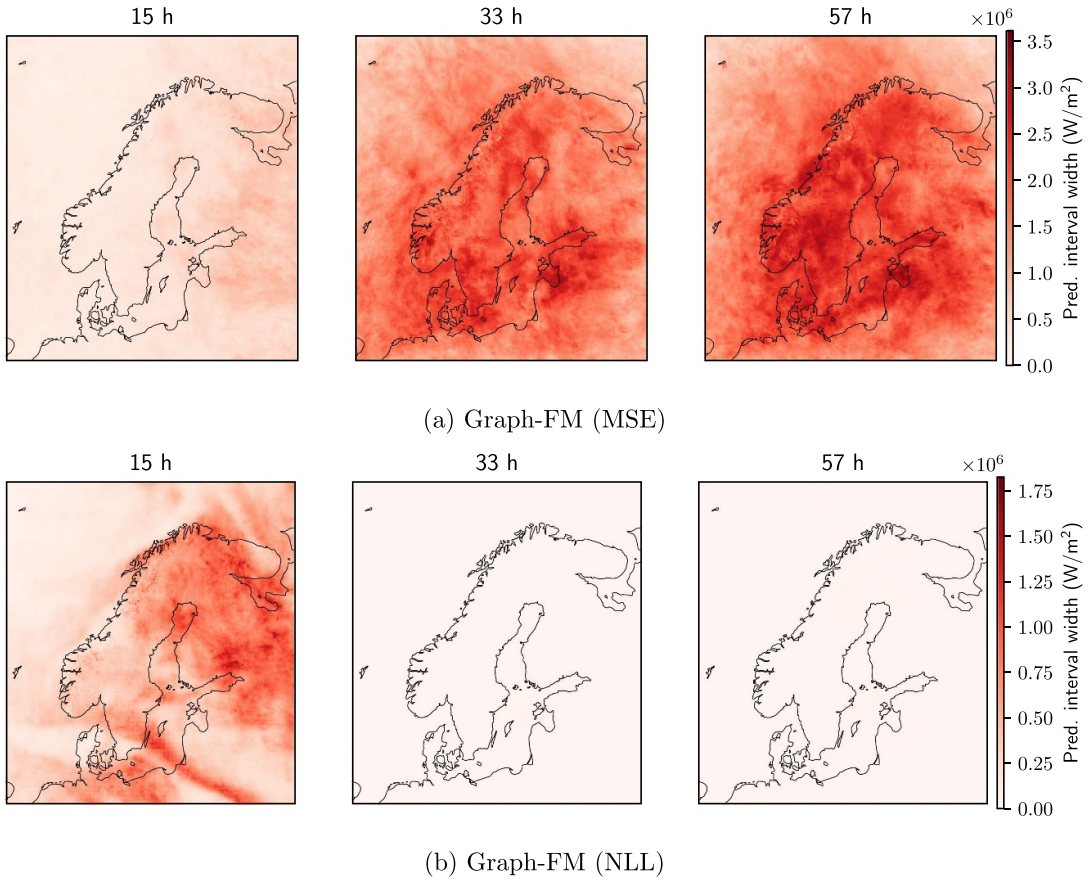


Figure 15. Width of predictive interval at $\alpha = 0.05$ for shortwave solar radiation (`nswrs`) in an example forecasts. Note that for Graph-FM (MSE) these widths are constant after calibration, while for Graph-FM (NLL) they depend on the predicted standard deviations for the specific forecast. This is most noticeable by Graph-FM (NLL) having very tight bounds during the night (lead times 33 h and 57 h, when the short-wave solar radiation is close to 0 and easy to predict.

by ERA5 at each forecasted time point. Using a full year for calibration allows for capturing the model performance across all different seasons. This allows for calibrating the model once, and then using the computed $\hat{q}$ values for the full next year of forecasts. However, any distributional shift due to climate variations from one year to the next remains. We note that in practic,e this does not seem to cause any major issue for achieving the desired coverage.

Figure 17 shows an example prediction from Graph-FM (MSE) and corresponding error bars. Global forecasting up to 10 days is a more challenging task than the limited area modelling up to 57 h. We see that at 10 days the model prediction fails to capture much of the patterns in the ground truth data. Importantly, this is accurately captured in the error bars, which increase with the lead time to high values at 10 days.

In figures 18 and 19 we plot the width of the error bar for specific humidity and wind. Similar to the limited area case, we note for Graph-FM (NLL) the error bars corresponds to patterns in the forecast itself, due to the use of predicted STDs from the model. For Graph-FM (MSE) the plots instead highlight the regions where predictions are more challenging in general, across all forecasts. Additional plots from the weather forecasting experiments are given in appendix H.

As for all experiments, we include results for the weather forecasting models in table 1. In both the global and limited area setting CP successfully produces calibrated error bars. For the Graph-FM (NLL) model the original STDs output by the model are too low, leading to invalid error bars and insufficient coverage. After applying CP however the error bars are well calibrated. We generally see that the Graph-FM (NLL) has tighter error bars than Graph-FM (MSE). This can be attributed to these being input-dependent, specific to each forecast from the model.

3.7.4. Discussion on exchangeability

Traditionally, the CP framework is limited in application to time-series modelling as it fails the exchangeability assumption. Previous research has looked into fixing this violation of exchangeability

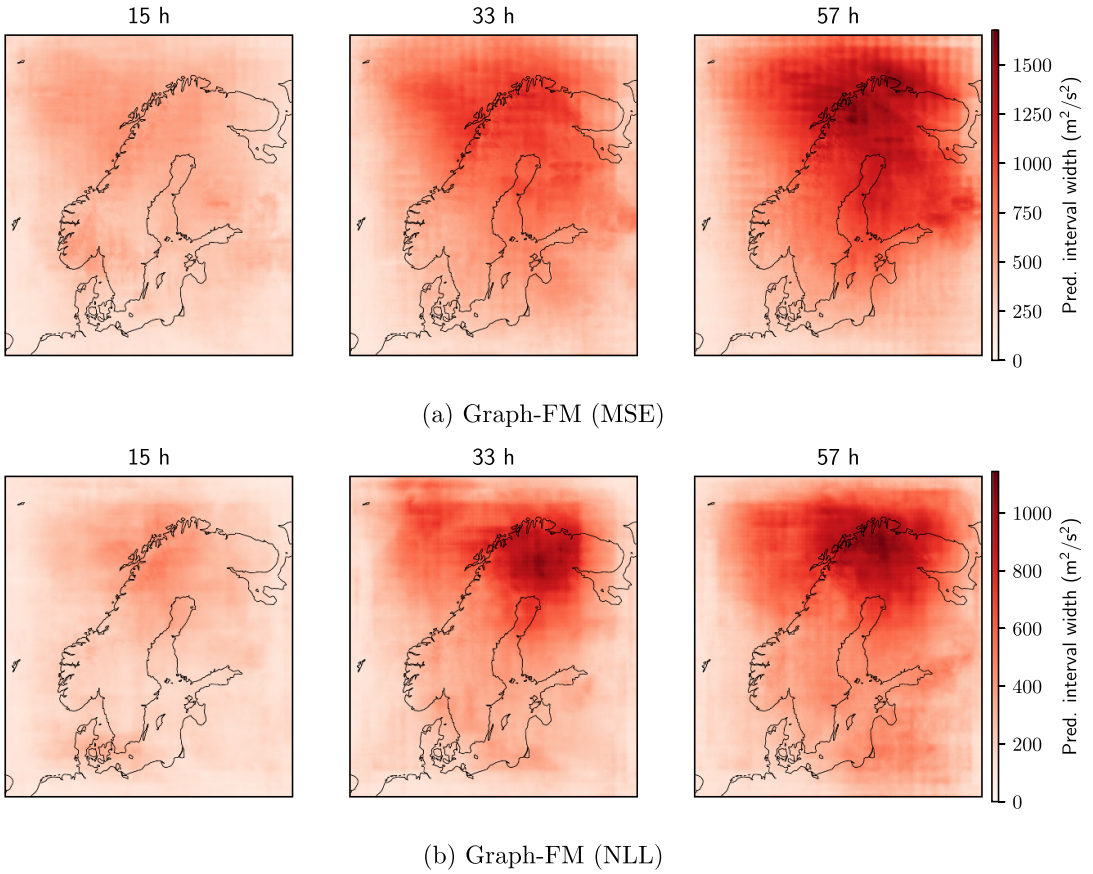


Figure 16. Width of predictive interval at $\alpha = 0.05$ for geopotential at 500 hPa (`z500`). Both models show a certain spatial pattern, especially for longer lead times. This pattern can be connected to how the GNN in Graph-FM is defined over the forecasting area.

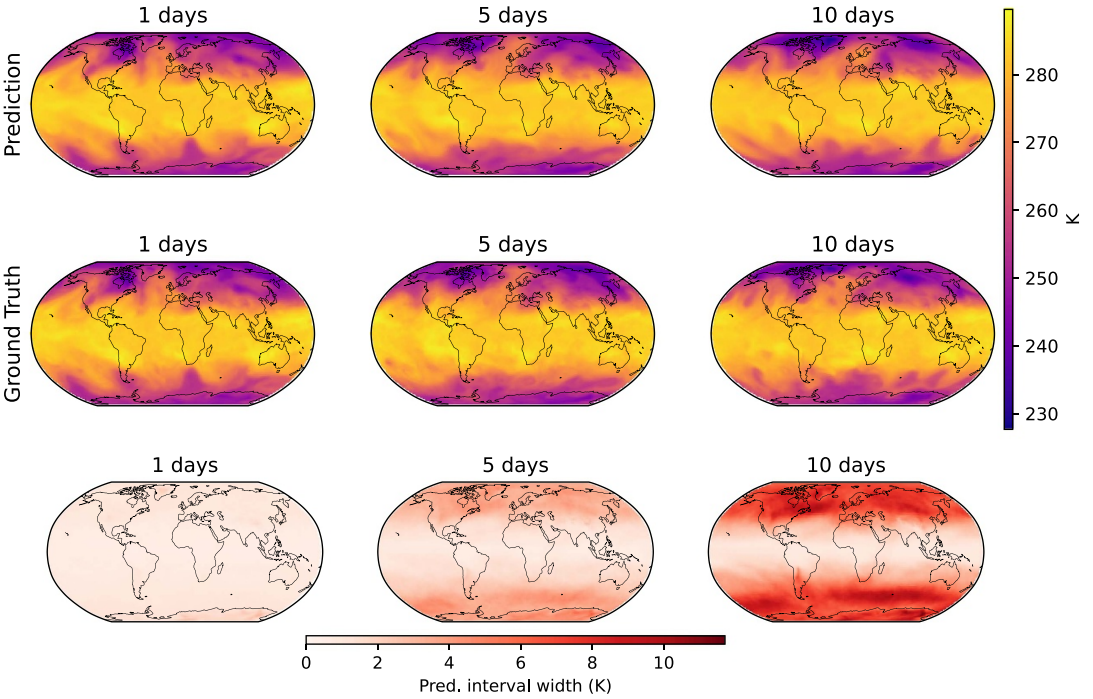
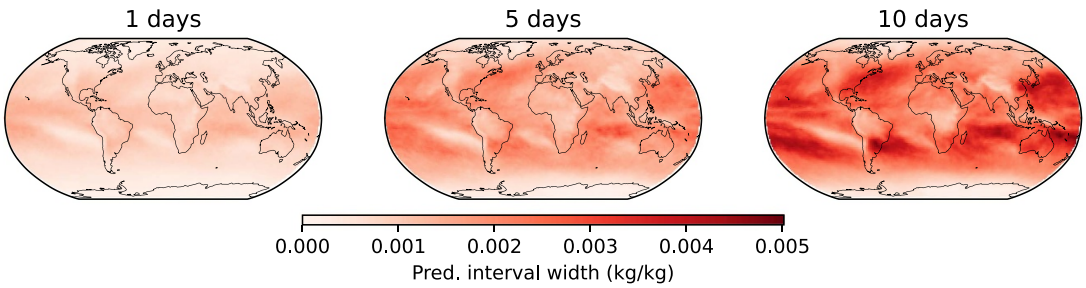
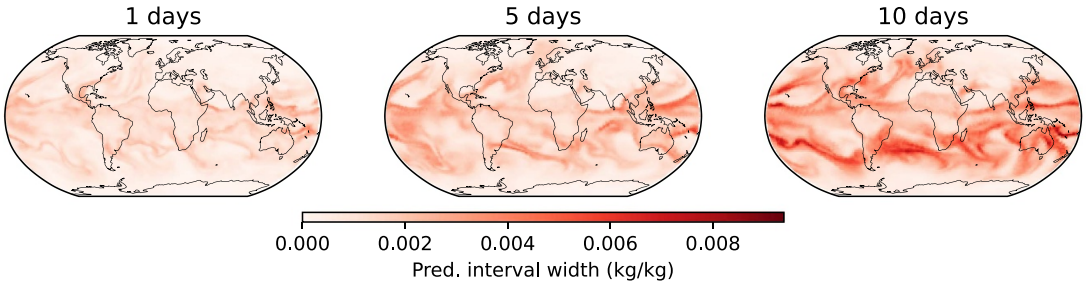


Figure 17. Prediction (top), Ground Truth (middle) and width of the error bars (bottom) at $\alpha = 0.15$ for predicting the temperature at 700 hPa (`t700`) using Graph-FM (MSE).

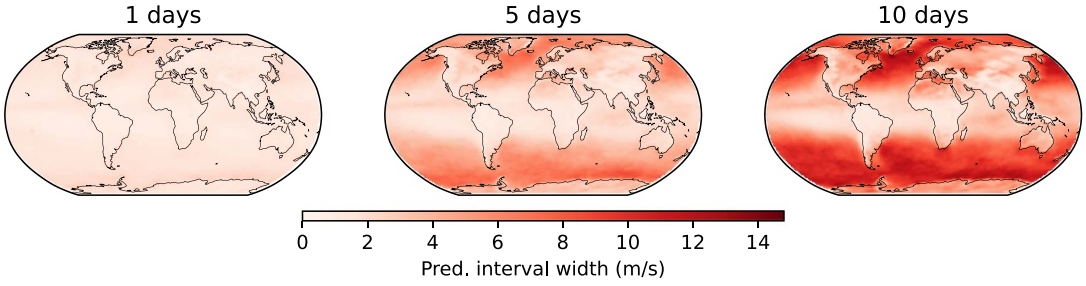


(a) Graph-FM (MSE)

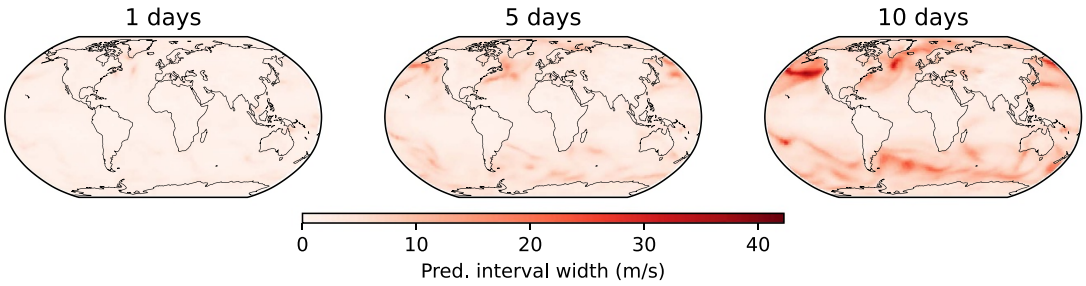


(b) Graph-FM (NLL)

Figure 18. Width of predictive interval at $\alpha = 0.15$ for specific humidity at 700 hPa (q600).



(a) Graph-FM (MSE)



(b) Graph-FM (NLL)

Figure 19. Width of predictive interval at $\alpha = 0.15$ for u-component of wind at 10 m above ground (10 u).

by accounting for the distribution shift using weighted conformal techniques (Tibshirani *et al* 2019), but becomes limited in application in multi-variate settings. Other work has explored CP for multi-variate time series forecasting, where each time-series is treated as an exchangeable observation (Stankeviciute *et al* 2021). Within the weather modelling tasks outlined in this section, we maintain exchangeability by

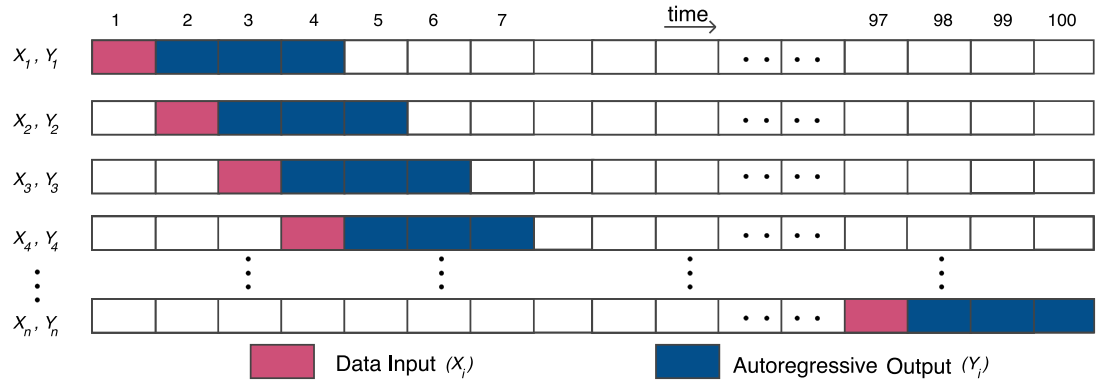


Figure 20. Constructing exchangeable input–output pairs from a time series dataset. The calibration dataset is arbitrarily characterised by a spatio-temporal dataset spanning 100 time instances, where the model takes in the field at a day and autoregressively predicts for the next 3 time instances. Since the model represents an initial boundary value problem, we are able to break each input–output (X_i, Y_i) as an exchangeable data point, essentially sampled from a distribution spanning the entire month.

treating each modelling task as an initial boundary value problem (IBVP, boundary given by the forcing terms in equation (5)). As given in equation (5), the model takes in the initial conditions, $\mathcal{X} \in \mathbb{R}^{T_{\text{in}=1} \times N_x \times N_y \times N_{\text{var}}}$ and is auto-regressively rolled out T_{out} steps to obtain the output $\mathcal{Y} \in \mathbb{R}^{T_{\text{out}} \times N_x \times N_y \times N_{\text{var}}}$. Being dependent on the initial conditions alone and being rolled out for a fixed number of steps, each input–output pair as mathematically outlined in section 2.1.1 and visually represented in figure 20 can be treated as an exchangeable pair. We are allowed to make this assumption on exchangeability since the model is agnostic to the temporal nature of the dataset beyond the autoregressive roll-out of each forward prediction, typical of an initial boundary value problem i.e. the neural weather forecast starting from 18:00 3*rd* January is independent of the forecast made using the neural weather models starting at 12:00 1*st* January. Here the calibration dataset is seen as samples from an extremely large distribution which effectively characterises the entirety of the entire month/year under consideration. Thus, by combining our preservation of spatio-temporal structure as outlined in section 2.1.1 and by treating the neural weather models as initial boundary value problems, we are able to maintain exchangeability across the calibration datasets, allowing us to perform CP.

Though the above description discusses about exchangeability across the calibration dataset, it does not extend across to the prediction set. For each of the experiments within the limited area and global weather forecasting, we assume that the climate does not vary significantly across the years under consideration for the calibration and prediction sets.

3.8. Camera diagnostic on a tokamak

The Mega-Ampere Spherical Tokamak (MAST) at the UK Atomic Energy Authority was equipped with fast Photron camera diagnostics to capture plasma evolution in the visible spectrum in real-time. These cameras have been instrumental in understanding plasma phenomena (Kirk *et al* 2006), providing statistical insights into plasma turbulence (Walkden *et al* 2022) and disruptions (Ham *et al* 2022).

Building on our previous work (Gopakumar *et al* 2024), we apply the CP framework to an FNO trained to forecast plasma evolution from camera imagery. The model takes 10 consecutive camera frames as input and predicts the subsequent 10 frames. Using the AER nonconformity score, we demonstrate that CP provides statistically valid error bars for these predictions. Details on the camera data, FNO architecture, and training are available in Gopakumar *et al* (2024).

3.8.1. Exchangeability and coverage validation

Although the FNO predicts plasma evolution over an entire shot duration, we structure the problem as an initial value problem where each forecast depends solely on its initial 10 frames. As illustrated in figure 20, this allows us to treat each input–output pair as exchangeable: the model predicts the entire spatio-temporal output tensor simultaneously rather than sequentially, preserving the temporal structure.

The validity of CP for this application rests on three key properties: (1) only the calibration dataset and prediction X_{n+1} must be exchangeable, (2) we predict error bars over the complete time interval simultaneously, not autoregressively, and (3) we assume minimal distributional shift between calibration

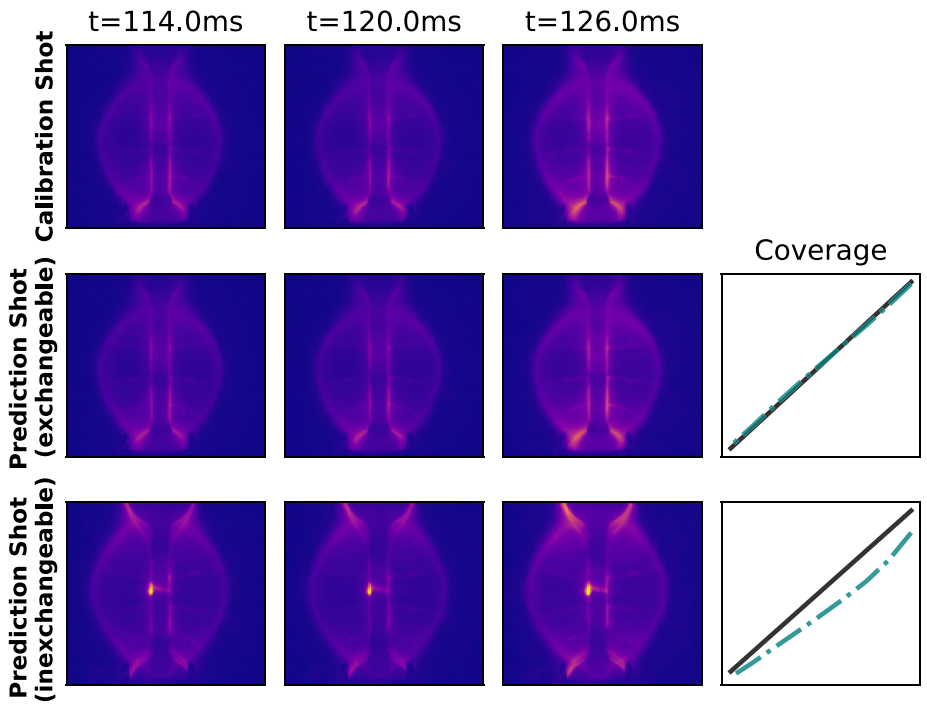


Figure 21. Coverage comparison for exchangeable (similar plasma profiles) versus non-exchangeable (dissimilar plasma profiles) prediction shots relative to the calibration set. Violations of exchangeability lead to insufficient coverage.

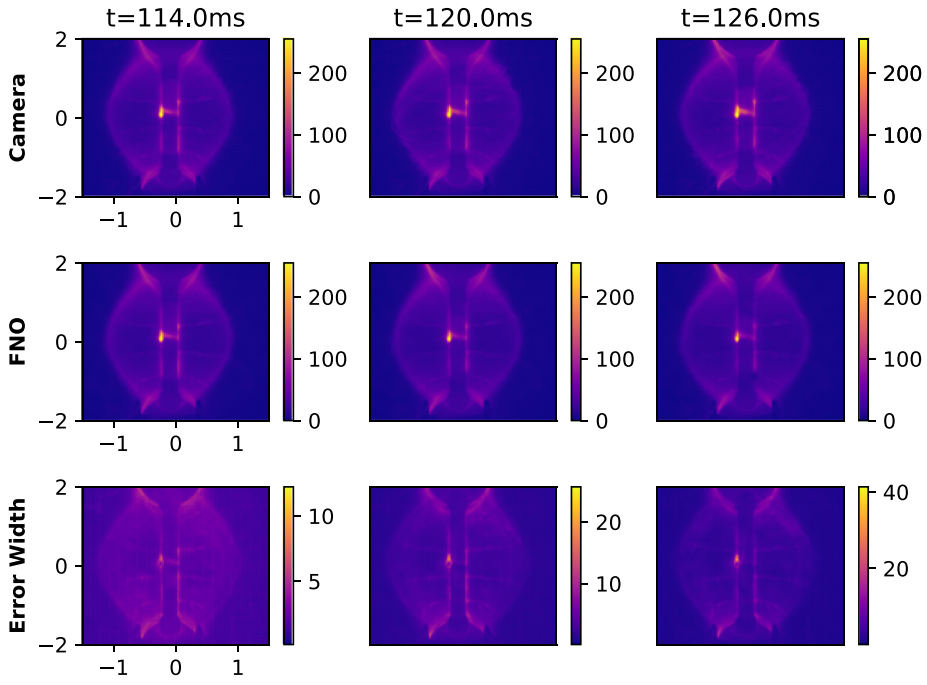


Figure 22. Camera observations (top), FNO predictions (middle), and width of prediction intervals obtained using CP with $\alpha = 0.5$ corresponding to 50% coverage (bottom).

and prediction shots. This final assumption requires that calibration and prediction shots exhibit similar plasma discharge profiles and device conditions.

To assess sensitivity to exchangeability violations, we compare coverage across shots with similar versus dissimilar plasma profiles relative to the calibration set. Figure 21 demonstrates that predictions on shots with substantially different characteristics suffer from insufficient coverage, highlighting the importance of the exchangeability assumption across the calibration and prediction set. The calibration

shot is indicative of a L-mode of confinement within the plasma, where as the inexchangeable prediction shot shows H-mode of confinement characterising a vastly different physics regime (Howlett *et al* 2023). When this assumption holds, CP provides exact coverage as shown in appendix I. Figure 22 visualises the calibrated error bars for plasma evolution forecasts across the tokamak central solenoid, demonstrating 50% coverage ($\alpha = 0.5$).

4. Discussion

We have demonstrated that CP provides a practical, theoretically grounded approach to uncertainty quantification for surrogate models across diverse spatio-temporal applications. Through comprehensive empirical evaluation spanning PDEs, fusion diagnostics, and weather forecasting, CP delivers statistically guaranteed marginal coverage regardless of model architecture, training regime, or output dimensionality. However, understanding both its capabilities and inherent limitations is essential for responsible application in scientific domains. Below, we discuss CP's key strengths for practical deployment, followed by an honest assessment of its limitations and implications for real-world use cases

4.1. Strengths

In safety-critical applications such as fusion reactor design, climate modelling, and engineering optimisation, surrogate models must provide credible uncertainty estimates alongside their predictions (Begoli *et al* 2019). CP addresses this need by offering statistical guarantees for uncertainty quantification with several key advantages:

Statistical guarantees. CP provides provable marginal coverage (equation (1)) regardless of model architecture, training regime, or output dimensionality. This validity is particularly crucial when surrogate models transition from research to production environments where retraining opportunities are limited.

Model-agnostic and scalable. The framework requires no architectural modifications or knowledge of training procedures, enabling application to pre-trained models. Our experiments demonstrate guaranteed coverage across outputs spanning up to 20 million dimensions (weather forecasting) with near-zero calibration costs, effectively circumventing the curse of dimensionality.

Computational efficiency. Unlike ensemble methods or Bayesian approaches requiring extensive sampling, CP calibration is computationally trivial (section 2.3). Calibration times range from seconds for low-dimensional problems to minutes for high-dimensional applications, performed on standard hardware without specialised computational resources.

Practical utility. CP enables rigorous validation of a surrogate model's usefulness for specific downstream applications, providing actionable uncertainty estimates for decision-making in risk-averse scenarios.

4.2. Limitations

While CP offers substantial benefits, several inherent limitations must be acknowledged for responsible application in scientific domains. We discuss these limitations, their practical implications, and potential mitigation strategies.

Marginal vs conditional coverage. The coverage guarantee in equation (1) provides *marginal coverage*-validity averaged over all predictions-rather than the more desirable *conditional coverage*:

$$\mathbb{P}(Y_{n+1} \in \mathbb{C}^{\alpha} | X_{n+1}) \geqslant 1 - \alpha. \tag{6}$$

In practice, this means that while 90% of predictions will be covered on average, any individual prediction may have substantially different actual coverage. Although conditional coverage cannot be guaranteed in general, approximations exist (Vovk 2012).

Our formulation guarantees coverage cell-wise across the spatio-temporal tensor but does not provide joint coverage across the entire prediction domain. Extensions to joint coverage exist

(Diquigiovanni *et al* 2021, Messoudi *et al* 2021, 2022) but fail to scale with dimensionality, limiting their applicability to the high-dimensional problems considered here.

Practical impact: In risk-critical scenarios requiring reliable bounds for specific predictions (e.g. fusion disruption avoidance, extreme weather events), marginal coverage may be insufficient. Users must understand that individual predictions, particularly in distribution tails, may not achieve the target coverage. Our experiments on exchangeability violations (section 3.8 and figure 21) demonstrate this sensitivity.

Mitigation: Covariate shifting methods (Tibshirani *et al* 2019) can improve conditional coverage when probability densities of calibration and deployment distributions are known or estimable. However, density estimation becomes unreliable in high dimensions (Quionero-Candela *et al* 2009). For critical applications, we recommend conservative α values and validation on held-out data similar to deployment conditions.

Data requirements and exchangeability. CP requires calibration data that is exchangeable with the prediction regime. The quality of coverage guarantees follows a beta distribution (equation (4)) governed by calibration set size. Our empirical study (appendix G) shows that $n_{\text{cal}} \geqslant 1000$ typically provides reliable coverage.

Practical impact: For experimental data (weather, fusion diagnostics), obtaining sufficient exchangeable calibration data can be challenging. The exchangeability assumption is particularly delicate for time-series data where distribution shifts are common. As demonstrated in figure 21, violations lead to invalid coverage. Weather experiments (section 3.7) assume minimal climate variation between calibration (2021) and prediction (2022) years-an assumption that may not hold for long-term climate change scenarios or extreme events outside the calibration distribution.

Mitigation: When calibration data is expensive (e.g. fusion experiments), fine-tuning scenarios offer a natural solution: training data doubles as calibration data since both represent the target distribution. For time-series applications, careful validation of exchangeability assumptions and sensitivity analyses (as performed for camera diagnostics) is essential.

Prediction sets vs distributions. CP provides prediction *sets* (intervals) rather than full probability distributions. While Bayesian methods offer distributions that can be propagated via Monte Carlo sampling or used for risk calculations, CP intervals lack this flexibility.

Recent work (Cella and Martin 2022) provides imprecise probabilistic interpretations of CP, enabling uncertainty propagation (Balch 2012, Hose and Hanss 2021). However, these methods are not yet widely adopted. It is worth noting that while Bayesian posteriors appear more informative, the false-confidence theorem (Martin 2019) shows that precise probability models can assign high confidence to low-probability events.

Input independence in deterministic models. For deterministic models using AER nonconformity scores, the error bars ($\sim \hat{q}$) are fixed during calibration and do not vary with inputs. As shown in figure 12, this produces global error estimates rather than input-specific bounds. This limitation is less severe for probabilistic models (STD scores) where prediction sets scale with model-predicted STDs (figure 2), providing a weak conditionining on the input.

Practical impact: Deterministic CP may produce overly conservative bounds in easy regions and insufficient bounds in challenging regions. This is particularly evident in foundation models trained on diverse data, where local complexity varies substantially.

Mitigation: When possible, use probabilistic models or ensemble approaches that provide initial uncertainty estimates for STD-based CP. Alternatively, recent work on normalised CP (Johansson *et al* 2021)

offers more input-adaptive bounds, though at the cost of increased computational complexity or architectural modifications.

Spatial and temporal correlations. Our cell-wise calibration treats each spatio-temporal point independently, ignoring correlations between adjacent cells. This is a significant limitation for PDE-based systems where solutions exhibit strong spatial and temporal dependencies.

Practical impact: While we achieve marginal coverage at each cell, the framework does not capture or leverage the correlation structure inherent in physical systems. This may lead to overly conservative joint coverage across regions or miss spatially coherent error patterns.

Mitigation: We implicitly rely on the surrogate model to learn spatial dependencies during training. Future extensions could incorporate spatial or temporal correlation structures into the CP framework, leading to joint coverage as we have explored in Gopakumar *et al* (2025).

4.3. Broader context

Despite these limitations, CP offers a valuable and practical tool for UQ in scientific machine learning. The guaranteed marginal coverage, model-agnostic nature, and computational efficiency make it particularly suitable for validating pre-trained surrogate models before deployment. When combined with careful validation of assumptions (particularly exchangeability), CP provides actionable uncertainty quantification that can guide decision-making in safety-critical applications.

For optimal results, we recommend: (1) validating exchangeability assumptions through sensitivity analyses, (2) using conservative α values for risk-critical applications, (3) preferring probabilistic models or ensembles when input-dependent bounds are crucial, and (4) maintaining awareness of the marginal nature of coverage guarantees when interpreting individual predictions.

5. Conclusion

This paper presents a comprehensive empirical study demonstrating that CP provides statistically guaranteed uncertainty quantification for surrogate models across diverse scientific applications. By maintaining exchangeability of spatio-temporal data through preservation of tensorial structure, we achieve valid error bars satisfying equation (1) for outputs spanning up to 20 million dimensions.

Key contributions. Our work establishes that CP can be applied to any pre-trained or fine-tuned surrogate model-regardless of architecture (MLP, U-Net, FNO, ViT, GNN), training regime, or output dimensionality-to obtain guaranteed marginal coverage with near-zero computational cost. We benchmark three nonconformity scores (CQR, AER, STD) across both deterministic and probabilistic models, demonstrating consistent coverage across applications ranging from fundamental PDEs to operational weather forecasting and fusion diagnostics.

Critically, we show that CP provides valid prediction sets even for out-of-distribution scenarios where models are deployed on physics regimes different from their training distribution (wave equation at half-speed, Navier–Stokes at different viscosity, pre-trained foundation models on new physics). This capability is essential for validating surrogate model utility in production environments where retraining is infeasible.

Practical impact. For scientific machine learning practitioners, our framework offers a rigorous method to assess whether a pre-trained model is suitable for a specific downstream application. The guaranteed coverage enables confident deployment of surrogate models in safety-critical contexts-from fusion reactor control to extreme weather response-where uncertainty quantification is imperative but computational budgets preclude ensemble methods or extensive Bayesian inference.

Scope and limitations. While we focus on spatio-temporal data, our methodology extends to any models producing fixed tensorial outputs with exchangeable calibration and prediction regimes. However, users must carefully validate exchangeability assumptions, particularly for time-series and experimental data where distribution shifts are common. As discussed in section 4, the marginal nature of coverage guarantees, cell-wise independence assumptions, and input independence in deterministic models represent important limitations that practitioners should consider when applying CP to their specific problems.

Future directions. Extensions to conditional coverage, incorporation of spatial–temporal correlation structures, and methods for handling systematic exchangeability violations remain important open problems. The integration of CP with recent advances in foundation models for scientific computing presents particularly promising opportunities for scalable, trustworthy uncertainty quantification.

Reproducibility. All code, data generation scripts, and trained models are publicly available at https://github.com/gitvicky/Spatio-Temporal-UQ.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/gitvicky/Spatio-Temporal-CP.

Acknowledgment

The authors would like to thank Anima Anandkumar and Zongyi Li from Caltech for their help with defining neural operators and extending them to complex physics cases. The authors would also like to thank Michael McCabe at the Flatiron Institute for his help in setting up the Multi-Physics Pretrained foundation physics model and extending it to a Fusion-relevant database. This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council. This work was supported under project ID a122 as part of the Swiss AI Initiative, through a grant from the ETH Domain and computational resources provided by the Swiss National Supercomputing Centre (CSCS) under the Alps infrastructure. It was also supported by the Excellence Center at Linköping–Lund in Information Technology (ELLIIT). Computations were enabled by the Berzelius resource at the National Supercomputer Centre, provided by the Knut and Alice Wallenberg Foundation. This work has been (part-) funded by the EPSRC Energy Programme [Grant Number EP/W006839/1]. To obtain further information on the data and models underlying this paper, please contact PublicationsManager@ukaea.uk*.

Author contributions

Vignesh Gopakumar 0000-0003-0904-3448
Conceptualization (equal), Data curation (equal), Formal analysis (equal), Investigation (equal), Methodology (equal), Visualization (equal), Writing – original draft (equal)

Ander Gray
Conceptualization (equal), Data curation (equal), Formal analysis (equal), Investigation (equal), Methodology (equal), Validation (equal), Visualization (equal), Writing – original draft (equal), Writing – review & editing (equal)

Joel Oskarsson 0000-0002-8201-0282
Data curation (equal), Investigation (equal), Resources (equal), Software (equal), Visualization (equal), Writing – original draft (equal)

Lorenzo Zanisi
Validation (equal), Writing – review & editing (equal)

Daniel Giles
Project administration (equal), Supervision (equal), Validation (equal), Writing – review & editing (equal)

Matt J Kusner
Supervision (equal), Writing – review & editing (equal)

Stanislas Pamela 0000-0001-8854-1749
Supervision (equal), Validation (equal), Writing – review & editing (equal)

Marc Peter Deisenroth
Formal analysis (equal), Project administration (equal), Supervision (equal), Validation (equal), Writing – review & editing (equal)

Appendix A. Poisson equation

The Poisson equation in one-dimension takes the form:

$$\frac{\partial^2 u}{\partial x^2} = \rho, \quad x \in [0,1], \tag{7}$$

where u defines the field value, x the spatial domain, and ρ the density of the source.

The Poisson equation is solved with a finite difference scheme using the *py-pde* python package (Zwicker 2020). Equation (7) is constructed as an initial-value problem, where a scalar uniform field is initialised across the domain and evolved until convergence. A dataset comprising different instances of the 1D Poisson equation is constructed by sampling for different initial values uniformly from within the domain: $u_{\text{init}} \in [0,4)$.

A total of 7000 data points are generated, where 5000 are used to train an MLP with 3 layers and 64 neurons in each layer, 1000 are used to perform the calibration required to estimate the nonconformity scores and another 1000 for validation. Being a steady-state problem, the MLP learns how a scalar field evolves under the influence of the Laplacian, mapping from the initial to the final state of evolution. The network learns to map the initial condition to the final steady-state solution.

Each MLP is trained to take in the scalar initial field along the 32-point spatial domain to output the final field at the steady state. For the case of STD, the architecture is modified with 1D dropout layers. Each model is trained for up to 1000 epochs using the Adam optimiser Kingma and Ba (2015) with a step-decaying learning rate. The learning rate is initially set to 0.005 and scheduled to decrease by half after every 100 epochs. The model was trained using a quantile loss for the case of CQR and MSE loss in all other cases.

Appendix B. Convection–diffusion equation

B.1. Physics

Consider a modified version of the one-dimensional convection–diffusion equation used to model the transport of a fluid:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + u\frac{\partial D}{\partial x} - c\frac{\partial u}{\partial x}, \quad x \in [0,10],\ t \in [0,0.1] \tag{8}$$

$$u(x, t=0) = \exp\left(-\frac{(x-\mu^2)}{2\sigma^2}\right). \tag{9}$$

Here u defines the density of the fluid, x the spatial coordinate, t the temporal coordinate, D the diffusion coefficient, and c the convection velocity. The initial condition is parameterised by μ and σ^2, representing the mean and variance of a Gaussian distribution. The system is bounded by a no-flux boundary condition.

The numerical solution for the above equation is built using a Newtonian solver with a forward time centred space implementation in Python. We construct a dataset by Latin hypercube sampling across parameters D, c, μ, σ. Each parameter is sampled from within the domain given in table 2 to generate 3000 simulation points, each with its own initial condition, diffusion coefficient and convection velocity. We generate another 2000 data points, 1000 each for the calibration and procuring of the prediction sets. These datasets are built by sampling across a different domain of the diffusion coefficient and convection velocity, different from that used for training; see table 3 for details. We use a one-dimensional U-Net to model the evolution of the convection–diffusion equation. The U-Net learns to perform the mapping from the first 10 time instances to the next 10 time instances, learning across the different field parameters and initial conditions. A more detailed physics description and the training set-up of the model can be found in appendix B.

As discussed in section 3.2, the dataset is built by solving the one-dimensional Convection Diffusion equation numerically. The physics of the equation, given by the various coefficients, is sampled from a certain range as given in table 2. Each datapoint, as in each simulation, is generated with different Diffusion coefficients and wave velocities as described in section 3.2. Each simulation is run for 100 time iterations with a $\Delta t = 0.0005$ across a spatial domain spanning [0,10], uniformly discretised into 200 spatial units in the x-axis. Once the simulations are run and the dataset is generated, we downsample the temporal discretisation from 100 to 20 by taking every 5th time step. The sampling parameters governing the PDE solutions used for the training are given in table 2 and that used for the calibration and prediction is given in table 3.

Table 2. Domain range and sampling strategies across the coefficients and initial condition parameters for building the training dataset for the 1D convection–diffusion equation.

Parameter	Domain	Type
Diffusion coefficient (α)	$[\sin(\frac{x}{\pi}), \sin(\frac{x}{2\pi})]$	Continuous
Convection velocity (β)	$[0.1, 0.5]$	Continuous
Mean (μ)	$[1.0, 8.0]$	Continuous
Variance (γ)	$[0.25, 0.75]$	Continuous

Table 3. Domain range and sampling strategies across the coefficients and initial condition parameters for building the calibration and prediction datasets for the 1D convection–diffusion equation.

Parameter	Domain	Type
Diffusion coefficient (α)	$[\sin(\frac{x}{2\pi}), \sin(\frac{x}{4\pi})]$	Continuous
Convection velocity (β)	$[0.5, 1, 0]$	Continuous
Mean (μ)	$[1.0, 8.0]$	Continuous
Variance (γ)	$[0.25, 0.75]$	Continuous

Table 4. Architecture of the 1D U-Net deployed for modelling 1D convection–diffusion equation.

Part	Layer	Output Shape
Input	—	(50, 20, 200)
Encoder 1	`Conv1d/BatchNorm1d/Tanh`	(50, 32, 200)
Pool 1	`MaxPool1d`	(50, 32, 200)
Encoder 2	`Conv1d/BatchNorm1d/Tanh`	(50, 64, 100)
Pool 2	`MaxPool1d`	(50, 64, 100)
Encoder 3	`Conv1d/BatchNorm1d/Tanh`	(50, 128, 50)
Pool 3	`MaxPool1d`	(50, 128, 50)
Encoder 4	`Conv1d/BatchNorm1d/Tanh`	(50, 256, 25)
Pool 4	`MaxPool1d`	(50, 256, 25
Bottleneck	`Conv1d/BatchNorm1d/Tanh`	(50, 512, 12)
Decoder 4	`ConvTranspose1d/Encoder 4`	(50, 256, 25)
Decoder 3	`ConvTranspose1d/Encoder 3`	(50, 128, 50)
Decoder 2	`ConvTranspose1d/Encoder 2`	(50, 64, 100)
Decoder 1	`ConvTranspose1d/Encoder 1`	(50, 32, 200)
Rescale	`Conv1d`	(50, 10, 200)

B.2. Model and training

We train a U-Net to map the spatio-temporal evolution of the field variable, taking in the first 20 time instances ($T_$in) to the next 10 time instances ($T_$out). For the case of the Convection–Diffusion equation, we do not deploy an auto-regressive structure but perform a mapping from the initial distribution to the later distribution. The U-Net architecture can be found in table 4. For the case of STD, the architecture is modified with 1D dropout layers following each encoder and decoder of the U-Net. Though the values governing the evolution of Convection–Diffusion are relatively small, for better representation, we normalise the value with a linear range scaling, allowing the field values to lie between -1 and 1. Each model is trained for up to 500 epochs using the Adam optimiser (Kingma and Ba 2015) with a step-decaying learning rate. The learning rate is initially set to 0.005 and scheduled to decrease by half after every 100 epochs. The model was trained using a quantile loss for the case of CQR and an MSE loss in all other cases.

Appendix C. Wave equation

C.1. Physics

Consider the two-dimensional wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0, \quad x, y \in [-1, 1], \ t \in [0, 1] \tag{10}$$

$$u(x, y, t = 0) = \exp\left(-\alpha \left((x - \beta)^2 + (y - \gamma)^2 \right) \right) \tag{11}$$

$$\frac{\partial u(x, y, t = 0)}{\partial t} = 0, \qquad u(x, y, t) = 0, \quad x, y \in \partial\Omega, \ t \in [0, 1], \tag{12}$$

Table 5. Domain range and sampling strategies across the initial condition parameters for the 2D Wave equation. A 2D Gaussian peak with a given amplitude and position within the domain is sampled using a Latin hypercube.

Parameter	Domain	Type
Amplitude (α)	$[10, 50]$	Continuous
X position (β)	$[0.1, 0.5]$	Continuous
Y position (γ)	$[0.1, 0.5]$	Continuous

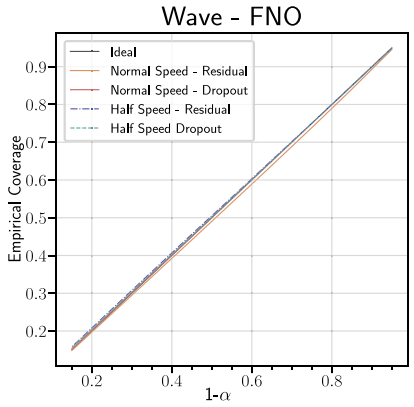


Figure 23. Coverage observed using AER and STD across datasets with normal wave speed (in-distribution) and half speed (out-distribution).

where u defines the field variable, c the wave velocity, x and y the spatial coordinates, t the temporal coordinates. α, β and γ are variables that parameterise the initial condition of the PDE setup. There exists an additional constraint to the PDE setup that initialises the velocity of the wave to 0. The system is bounded periodically within the mentioned domain.

The solution for the wave equation is obtained by deploying a spectral solver that uses a leapfrog method for time discretisation and a Chebyshev spectral method on a tensor product grid for spatial discretisation (Gopakumar *et al* 2023). The dataset is built by performing a Latin hypercube scan across the defined domain for the parameters α, β, γ, which accounts for the amplitude and the location of the Gaussian peak, sampled differently for each simulation. We generate 2500 simulation points, each one with its own initial condition and use 500 for training, 1000 each for calibration and procuring the prediction sets. We train a 2D U-Net and an FNO to learn the evolution of wave dynamics.

The physics of the equation, given by the various coefficients, is held constant across the dataset generation throughout, as given in equation (10). Each data point, as in each simulation, is generated with a different initial condition as described above. The parameters of the initial conditions are sampled from within the domain as given in table 5. Each simulation is run for 150-time iterations with a $\Delta t = 0.00667$ across a spatial domain spanning $[-1, 1]^2$, uniformly discretised into 33 spatial units in the x and y axes. Once the simulations are completed and the dataset is generated, we select the first 80 time instances of the evolution of each simulation to be used for training.

C.2. Model and training

We train U-Nets and FNOs to map the spatio-temporal evolution of the field variables. For the U-Nets, the network takes in the first 20 time instances (T_in) to map the next 30 time instances (*step*). The U-net performs a feed-forward mapping without any autoregressive roll-outs. For the FNO we deploy an auto-regressive structure that performs time rollouts, allowing us to map the initial time steps in a recursive manner up until the desired time instance (T_out). Each model autoregressively models the evolution of the field variable up until the 80th time instance. The U-Net architecture can be found in table 6 and the FNO in table 7. For the case of STD, the architecture is modified with 2D dropout layers following each encoder and decoder of the U-Net and after each Fourier layer within the FNO. We employ a linear range normalisation scheme, placing the field values between -1 and 1. Each model is trained for up to 500 epochs using the Adam optimiser (Kingma and Ba 2015) with a step decaying

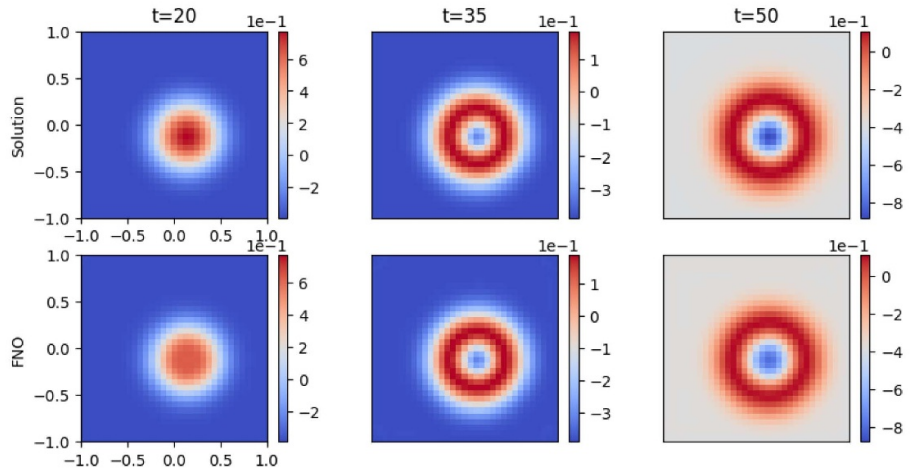


Figure 24. Waves: Temporal evolution of the field associated with the wave equation modelled using the numerical spectral solver (top of the figure) and that of the U-Net (bottom of the figure). The spatial domain is given in Cartesian geometry.

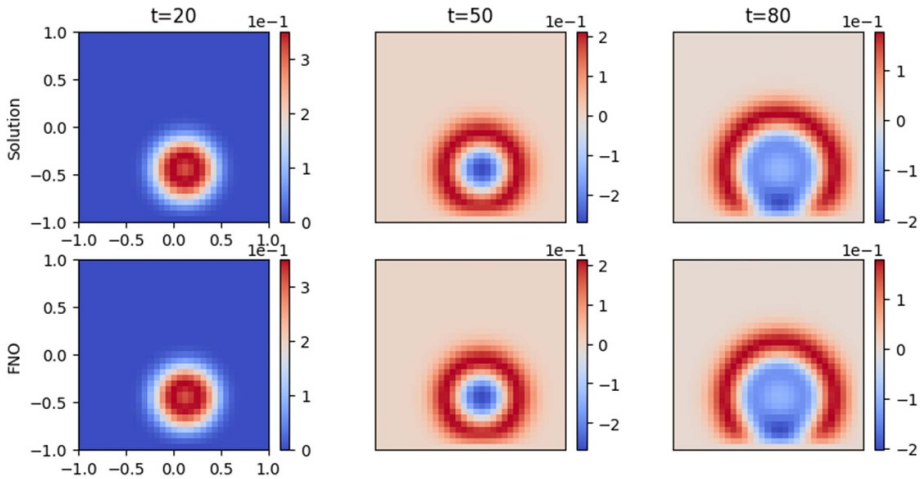


Figure 25. Waves: Temporal evolution of the field associated with the wave equation modelled using the numerical spectral solver (top of the figure) and that of the FNO (bottom of the figure). The spatial domain is given in Cartesian geometry.

Table 6. Architecture of the 2D U-Net deployed for the 2D Wave equation.

Part	Layer	Output shape
Input	—	(50, 20, 33, 33)
Encoder 1	`Conv2d/BatchNorm2d/Tanh`	(50, 32, 33, 33)
Pool 1	`MaxPool2d`	(50, 32, 33, 33)
Encoder 2	`Conv2d/BatchNorm2d/Tanh`	(50, 64, 16, 16)
Pool 2	`MaxPool2d`	(50, 64, 16, 16)
Bottleneck	`Conv2d/BatchNorm2d/Tanh`	(50, 128, 8, 8)
Decoder 2	`ConvTranspose2D/Encoder 2`	(50, 64, 16, 16)
Decoder 1	`ConvTranspose2D/Encoder 1`	(50, 32, 33, 33)
Rescale	`Conv2D`	(50, 10, 33, 33)

learning rate. The learning rate is initially set to 0.005 and scheduled to decrease by half after every 100 epochs. The model was trained using a quantile loss in the case of CQR and an MSE loss in the other cases.

Table 7. Architecture of the Individual FNO deployed for modelling the Wave equation.

Part	Layer	Output shape
Input	—	(50, 33, 33, 22)
Lifting	`Linear`	(50, 33, 33, 32)
Fourier 1	`Fourier2d/Conv2d/Add/GELU`	(50, 32, 33, 33)
Fourier 2	`Fourier2d/Conv2d/Add/GELU`	(50, 32, 33, 33)
Fourier 3	`Fourier2d/Conv2d/Add/GELU`	(50, 32 33, 33)
Fourier 4	`Fourier2d/Conv2d/Add/GELU`	(50, 32, 33, 33)
Fourier 5	`Fourier2d/Conv2d/Add/GELU`	(50, 32, 33, 33)
Fourier 6	`Fourier2d/Conv2d/Add/GELU`	(50, 32, 33, 33)
Projection 1	`Linear`	(50, 33, 33, 128)
Projection 2	`Linear`	(50, 33, 33, 10)

Appendix D. Navier–Stokes Equations for Vorticity

D.1. Physics

The Navier–Stokes scenario that we are interested in modelling is taken from the exact formulation in Li *et al* (2021), where the viscosity of the incompressible fluid in 2D is expressed as:

$$\frac{\partial w}{\partial t} + u\nabla w = \nu\nabla^2 w + f, \qquad x \in (0,1)\,,\ y \in (0,1)\,,\ t \in (0,T) \tag{13}$$

$$\nabla u = 0, \qquad x \in (0,1)\,,\ y \in (0,1)\,,\ t \in (0,T) \tag{14}$$

$$w = w_0, \qquad x \in (0,1)\,,\ y \in (0,1)\,,\ t = 0, \tag{15}$$

where u is the velocity field and vorticity is the curl of the velocity field $w = \nabla \times u$. The domain is split across the spatial domain characterised by x, y and the temporal domain t. The initial vorticity is given by the field w_0. The forcing function is given by f and is a function of the spatial domain in x, y. We utilise two datasets from Li *et al* (2021) that are built by solving the above equations with viscosities $\nu = 1e-3$ and $\nu = 1e-4$ under different initial vorticity distributions. For further information on the physics and the data generation, refer Li *et al* (2021).

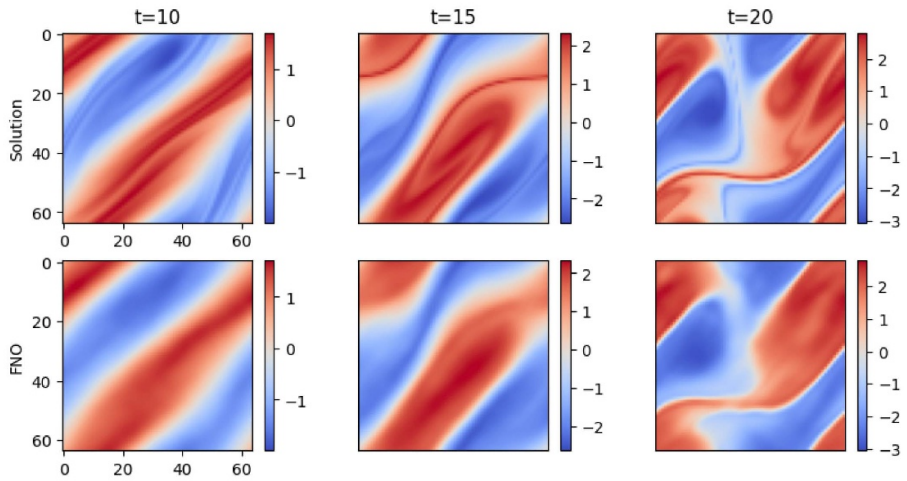
D.2. Model and training

We train an FNO to map the spatio-temporal evolution of the vorticity, taking in the first 10 time instances ($T_$in) to the next 10 time instances (*step*). For the case of the Navier–Stokes equations, we deploy a feed-forward mapping from the initial 10 time steps to the next 10 time steps. The architecture for the FNO can be found in table 8. We deploy a Min–Max normalisation strategy, allowing the field values to lie between -1 and 1. Each model is trained for up to 500 epochs using the Adam optimiser (Kingma and Ba 2015) with a step decaying learning rate. The learning rate is initially set to 0.005 and scheduled to decrease by half after every 100 epochs. The model was trained using a relative LP loss. Considering the efficiency and simplicity, CP for the Navier–Stokes case was conducted using AER and STD as a nonconformity metric as given in the section 2.2.

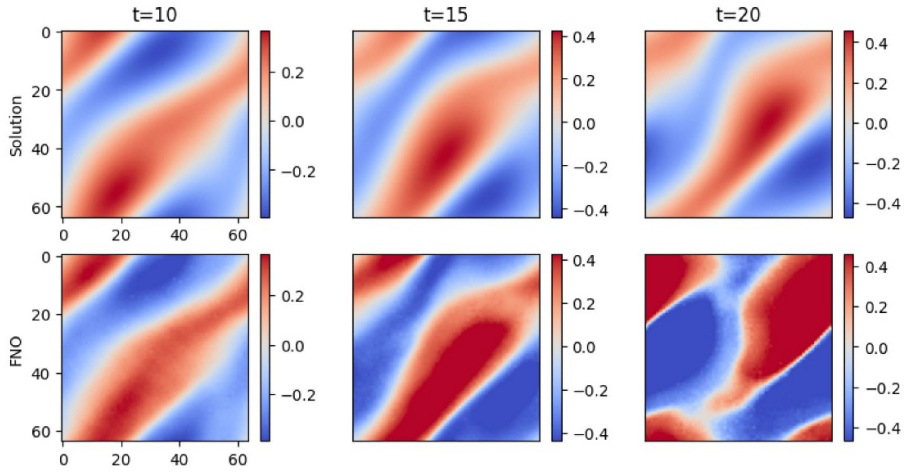
D.2.1. Prediction

Table 8. Architecture of the Individual FNO deployed for modelling the Navier–Stokes equation.

Part	Layer	Output shape
Input	—	(20, 64, 64, 12)
Lifting	`Linear`	(50, 64, 64, 32)
Fourier 1	`Fourier2d/Conv2d/Add/GELU`	(20, 16, 64, 64)
Fourier 2	`Fourier2d/Conv2d/Add/GELU`	(20, 16, 64, 64)
Fourier 3	`Fourier2d/Conv2d/Add/GELU`	(20, 16 64, 64)
Fourier 4	`Fourier2d/Conv2d/Add/GELU`	(20, 16, 64, 64)
Fourier 5	`Fourier2d/Conv2d/Add/GELU`	(20, 16, 64, 64)
Fourier 6	`Fourier2d/Conv2d/Add/GELU`	(20, 16, 64, 64)
Projection 1	`Linear`	(20, 64, 64, 128)
Projection 2	`Linear`	(20, 64, 64, 10)



(a) Training Distribution



(b) Out-of-distribution

Figure 26. Navier–Stokes: Temporal evolution of the vorticity associated with the Navier–Stokes equations. In (a), we compare the FNO performance against that of the numerical solver within the training distribution ($\nu = 1e-3$). In (b), we demonstrate the performance of the same FNO on out-of-distribution data upon which we perform conformal prediction ($\nu = 1e-4$) modelled using the numerical solver (top of the figure) and that of the FNO (bottom of the figure). The spatial domain is given in Cartesian geometry.

Appendix E. Magnetohydrodynamics of plasma blobs

E.1. Physics

The Reduced-MHD equations that we are interested in modelling can be described as:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \vec{v}) + D\nabla^2 \rho \tag{16}$$

$$\rho \frac{\partial \vec{v}}{\partial t} = -\rho \vec{v} \cdot \nabla \vec{v} - \nabla p + \mu \nabla^2 \vec{v} \tag{17}$$

$$\frac{\partial p}{\partial t} = -\vec{v} \cdot \nabla p - \gamma p \nabla \cdot \vec{v} + \kappa \nabla^2 T \tag{18}$$

where ρ is the density, p the pressure, T the temperature, and $\vec{v}$ the velocity. D is the diffusion coefficient, μ the viscosity, and κ the thermal conductivity. The ratio of specific heats γ is taken to be that of a monatomic gas, $\frac{5}{3}$.

Equation (16) depicts the continuity equation, modelling the evolution of density subject to diffusion, convection and the electrostatic potential. Equation (17) represents the conservation of momentum

Table 9. Domain range and sampling strategies across the initial condition parameters.

Parameter	Distribution	Type
Width	$U[0.02, 0.1]$	Continuous
Number of Blobs	$U[1, 10]$	Discrete
R—Position of Blobs	$U[9.4, 10.4]$	Continuous
Z—Position of Blobs	$U[-0.4, +0.4]$	Continuous
Amplitude of Density of Blobs	$U[0.5, 2.0]$	Continuous
Amplitude of Temperature of Blobs	$U[0.5, 3.0]$	Continuous

Table 10. Architecture of the multi-variable FNO deployed for modelling reduced MHD.

Part	Layer	Output shape
Input	—	(10, 3, 106, 106, 12)
Lifting	`Linear`	(10, 3, 106, 106, 32)
Fourier 1	`Fourier2d/Conv3d/Add/GELU`	(10, 3, 32, 106, 106)
Fourier 2	`Fourier2d/Conv3d/Add/GELU`	(10, 3, 32, 106, 106)
Fourier 3	`Fourier2d/Conv3d/Add/GELU`	(10, 3, 32, 106, 106)
Fourier 4	`Fourier2d/Conv3d/Add/GELU`	(10, 3, 32, 106, 106)
Fourier 5	`Fourier2d/Conv3d/Add/GELU`	(10, 3, 32, 106, 106)
Fourier 6	`Fourier2d/Conv3d/Add/GELU`	(10, 3, 32, 106, 106)
Projection 1	`Linear`	(10, 3, 106, 106, 128)
Projection 2	`Linear`	(10, 3, 106, 106, 5)

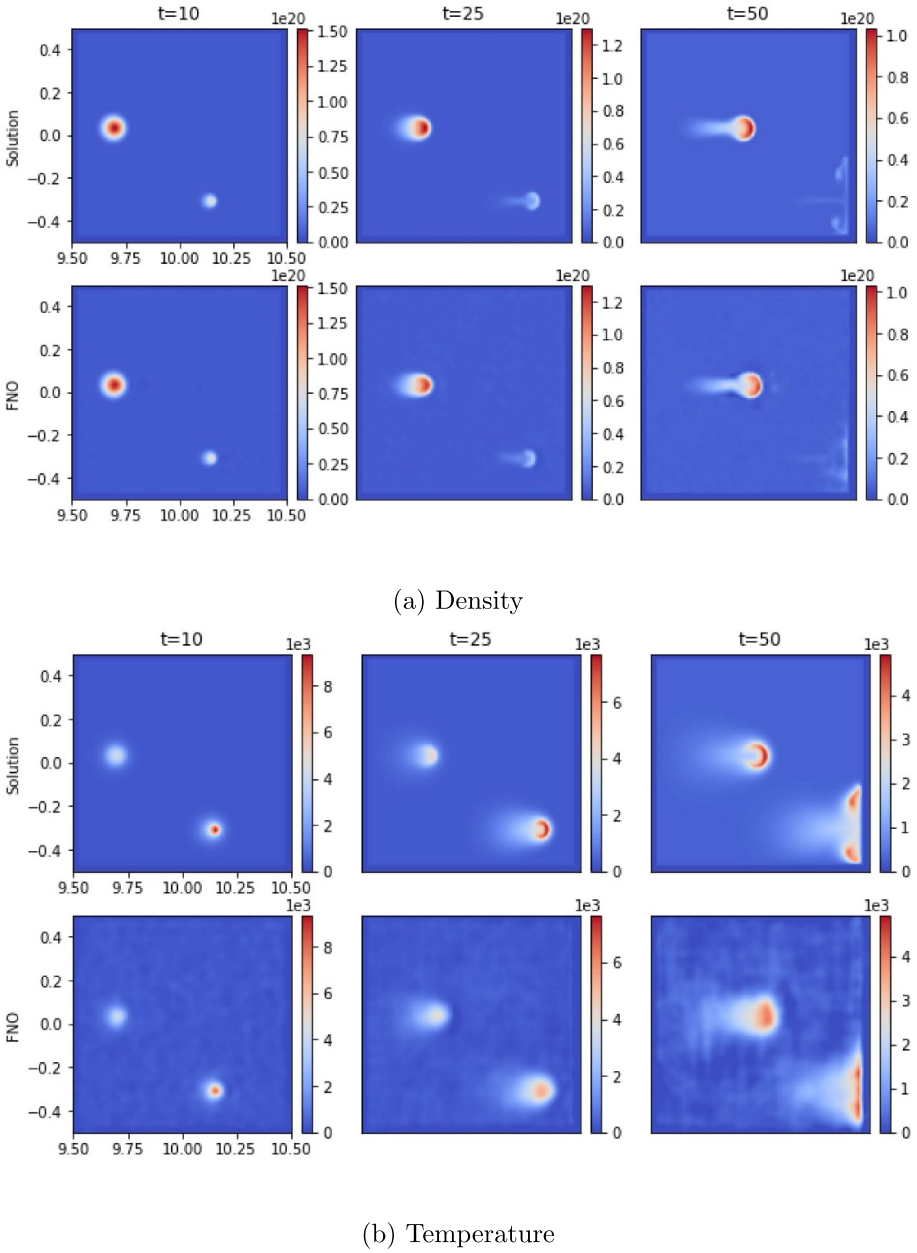
within the field. Equation (18) models the conservation of energy, characterised by the pressure, temperature and velocity.

Within each simulation, multiple-density blobs with varying positions, width and amplitude are initialised in a low-density background. In the absence of a plasma current to hold the density blob in place, the pressure gradient term in the momentum equation generates a buoyancy effect, causing the blob to move outwards. The system under consideration is characterised by a highly correlated multi-variable setting as given above. Within each simulation, we evolve the blobs to migrate radially outward until they reach the wall, where the Dirichlet boundary conditions engage to allow for convection and diffusion. Refer to Gopakumar *et al* (2024) for more detailed information about the setup.

E.2. Model and training

We train a multi-variable FNO to map the spatio-temporal evolution of the field variable, taking in the first 10 time instances ($T_$in) to the next 5 time instances (*step*). For the case of the MHD equations, we deploy an auto-regressive structure that performs a time rollout, allowing us to map the initial time steps in a recursive manner up until the desired time instance ($T_$out). Each model autoregressively models the evolution of the field variable up until the 50th time instance. The architecture for the multi-variable FNO can be found in table 10. We deploy a two-fold normalisation strategy considering the nature of the dataset. The physical field information represented within the MHD cases is in different scales, with densities ranging from 0 to $1e20$ and temperatures ranging up to $1e6$. Since we are considering the gradual diffusion of an inhomogeneous density blob(s), the data distribution within the spatial domain is severely imbalanced. Taking these aspects of the training data into consideration, a physics normalisation is performed initially, where the field values are scaled down by dividing by the prominent field value. This is followed up by a linear range scaling, allowing the field values to lie between -1 and 1. Each model is trained for up to 500 epochs using the Adam optimiser (Kingma and Ba 2015) with a step decaying learning rate. The learning rate is initially set to 0.005 and scheduled to decrease by half after every 100 epochs. The model was trained using a relative LP loss. Considering the efficiency and simplicity, CP for the MHD case was only conducted using the AER as a nonconformity metric, as given in the section 2.2.

E.3. Prediction

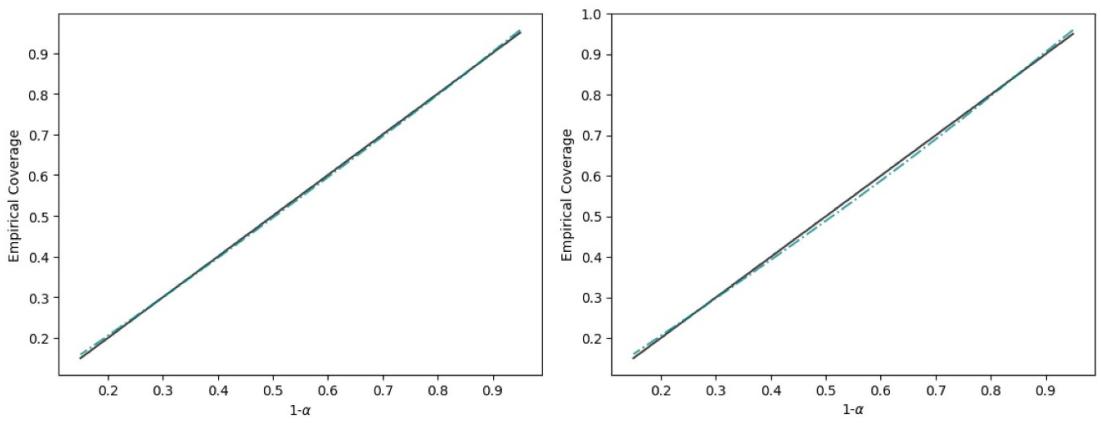


(a) Density

(b) Temperature

Figure 27. Multiple blobs: Temporal evolution of (a) the density and (b) the temperature variables describing the plasma evolution as obtained using the JOREK code (top of each image) and that of the multi-variable FNO (bottom of each figure). The spatial domain is given in toroidal geometry characterised by R in the x-axis and Z in the y-axis.

Appendix F. Empirical coverage of foundation physics models



(a) Pre-trained Model (b) Fine-tuned Model

Figure 28. Empirical coverage for Foundation Physics Models.

Appendix G. Impact of calibration size

Across almost all the experiments in section 3, we use a calibration dataset of the size 1000 exchangeable simulations. The only exception is for the multi-variable FNO for MHD, for which we use 100 simulations, as there was limited data available. We chose the number 1000 as the baseline size of the calibration dataset since in Angelopoulos and Bates (2023) they demonstrate that choosing $n_{\text{cal}} = 1000$ calibration points leads to a coverage that is typically between 0.88 and 0.92 for $\alpha = 0.1$. Since the size and nature of the calibration set are a source of finite sample variability, it requires analysis across each problem to which we deploy CP.

Ideally, equation (1) holds for a calibration dataset of any size n_{cal}. The coverage guaranteed by CP conditionally on this calibration dataset is essentially a random quantity. Thus, depending on the choice of the calibration dataset, the coverage would fluctuate around $1 - \alpha$. The distribution of the coverage as a function of the size of the calibration size is governed by a Beta distribution as given in equation (4).

We conduct an empirical study exploring the impact the size of the calibration dataset has on providing guaranteed coverage within our experiments. We iterate over $n_{\text{cal}} = 250, 500, 750, 1000$ for the Poisson (figure 29), Convection–Diffusion (figure 30) and the Wave equation (figures 31 and 32). Though the coverage obtained is from a Beta distribution governed by n_{cal} and changes with each sampling from that, our experiments are restricted to a single sample of n_{cal} data points from that distribution. The study of the impact of the calibration dataset is done across all the various nonconformity scores as well.

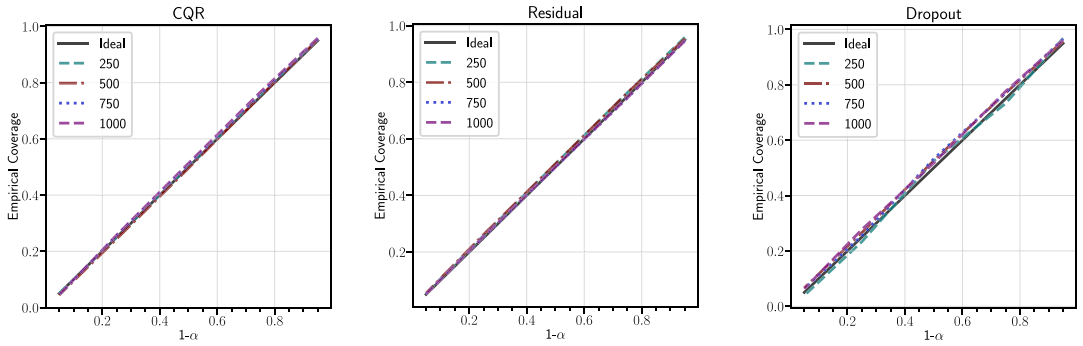


Figure 29. Impact of the size of the calibration dataset on the coverage obtained for conformal prediction across various nonconformity scores for the **Poisson equation**. Irrespective of the chosen size of the calibration dataset, we obtain guaranteed coverage.

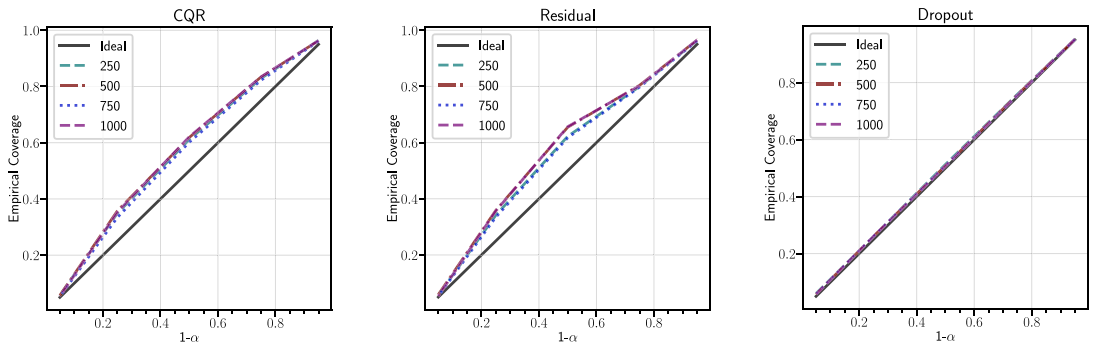


Figure 30. Impact of the size of the calibration dataset on the coverage obtained for conformal prediction across various nonconformity scores for the **Convection–Diffusion equation**. Irrespective of the chosen size of the calibration dataset, we obtain guaranteed coverage; however, with larger n_{cal} we obtain marginally better coverage. Irrespective of the chosen size of the calibration dataset, we obtain guaranteed coverage.

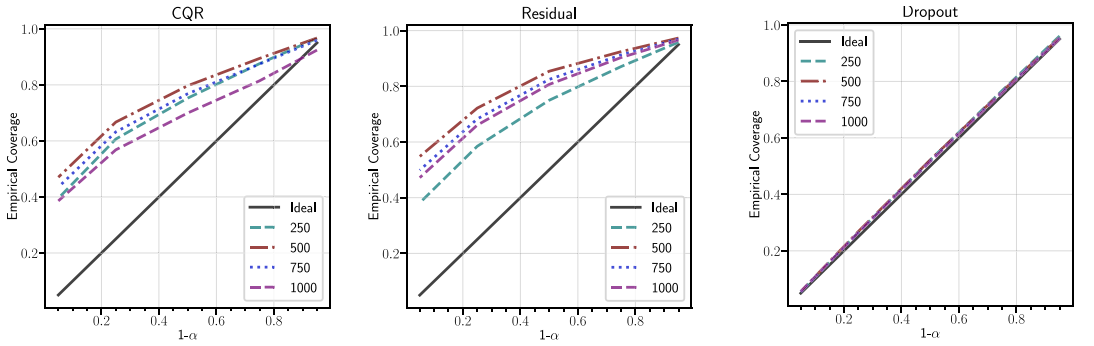


Figure 31. Impact of the size of the calibration dataset on the coverage obtained for conformal prediction across various nonconformity scores for the **Wave equation modelled using U-Net**. Irrespective of the chosen size of the calibration dataset, we obtain guaranteed coverage.

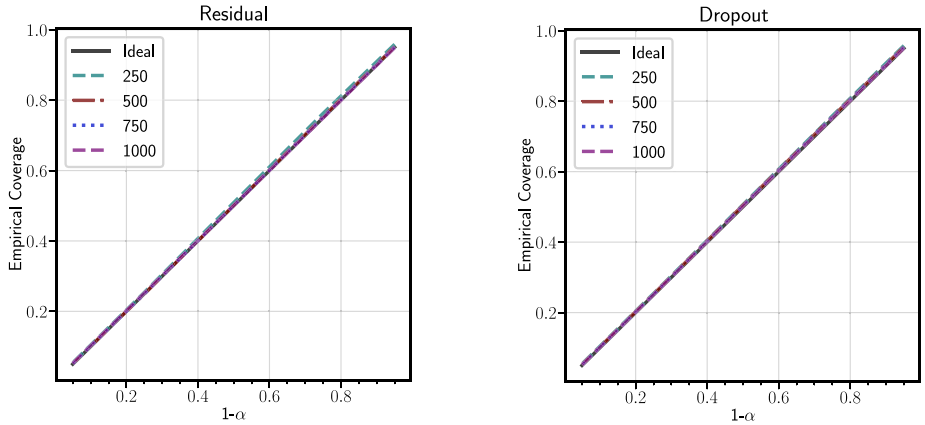


Figure 32. Impact of the size of the calibration dataset on the coverage obtained for conformal prediction across various nonconformity scores for the **Wave equation modelled using FNO**. Irrespective of the chosen size of the calibration dataset, we obtain guaranteed coverage.

From figures 29–32, we explore the impact of the size of the calibration dataset on the coverage obtained for the various nonconformity scores. We notice that across our experiments, we obtain guaranteed coverage, irrespective of the chosen nonconformity score or the size of the calibration dataset (n_{cal}). Though the size of the calibration dataset has no impact on the guarantee, as can be witnessed in the above figures, the tightness of the error bars is governed by the size of the calibration dataset.

Appendix H. Neural weather prediction models

H.1. Physics and data

H.1.1. Limited area forecasting—MetCoOp Ensemble Prediction System (MEPS) dataset

The MEPS provides operational weather forecasts for the Nordic region (Müller *et al* 2017). Our dataset comprises historical MEPS forecasts from April 2021 to March 2023, initialised at 00 and 12 UTC daily with 5 ensemble members per initialisation (*sim*10 forecasts per day).

Spatial configuration: The domain uses a Lambert conformal conic projection, downsampled from 2.5 km to 10 km resolution for computational efficiency, yielding a grid of $N_x \times N_y = 238 \times 268$ nodes.

Temporal configuration: Forecasts extend 66 hours with 1 hour steps. Models predict at 3 hour intervals using the last two states as input, resulting in an effective 57 hour evaluation horizon (19 time steps).

Variables: The dataset includes $N_{\text{var}} = 17$ variables: surface conditions (ground/sea level pressure, 2 m temperature and humidity), radiation fluxes (net longwave/shortwave), lowest atmospheric level (12.5 m: temperature, humidity, wind components), pressure levels (temperature at 500/850 hPa, wind at 850 hPa, geopotential at 500/1000 hPa), and integrated water vapour column. All variables except solar radiation are instantaneous.

Data splits: Training uses months 1–15 (April 2021–June 2022), validation uses months 16–18, calibration uses September 2021 forecasts (from 2021-09-04 onwards), and testing uses September 2022 forecasts. Using the same month for calibration and testing minimises seasonal distributional shifts.

H.1.2. Global forecasting—ERA5 dataset

Global models are trained on ERA5 reanalysis data (Hersbach *et al* 2020) at 1.5° resolution ($N_x \times N_y = 240 \times 121$ nodes). Forecasts extend 10 days with 6 hour steps ($T = 40$ time steps). The full dataset contains 83 variables (5 surface variables and 6 atmospheric variables at 13 pressure levels). For computational efficiency, CP experiments use $N_{\text{var}} = 11$ variables: all 5 surface variables and the 6 atmospheric variables at 700 hPa. Training uses multiple years of historical data, calibration uses the full year 2018, and testing uses 2019.

H.2. Model architecture and training

H.2.1. GNN architecture

The Graph-FM architecture (Oskarsson *et al* 2024) employs a hierarchical GNN with an encode-process-decode structure. Grid nodes represent the spatial discretisation, while mesh nodes form a coarser multi-resolution hierarchy ($L = 4$ levels for a limited area, $L = 8$ for global). Encoding maps grid states to the mesh, processing performs message passing across mesh levels (4 or 8 layers), and decoding projects back to the grid. All representations use 64-dimensional embeddings (limited area) or 256-dimensional embeddings (global).

H.2.2. Training procedure

Models undergo two-stage training: (1) single-step prediction (X^{t+1} from $X^{t-1:t}$), then (2) rollout fine-tuning with 4-step autoregressive sequences. Two model variants are trained:

Graph-FM (MSE): Uses weighted MSE loss for deterministic predictions:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N_{\text{rollout}}} \sum_t \frac{1}{|G|} \sum_{v \in G} \sum_{i=1}^{N_{\text{var}}} \lambda_i \omega_i \left(\hat{X}^t_{v,i} - X^t_{v,i} \right)^2 \tag{19}$$

where λ_i is the inverse variance of time differences, ω_i weights the vertical level, and G is the set of non-boundary grid nodes.

Graph-FM (NLL): Uses NLL loss with diagonal Gaussians, outputting mean μ and STD σ:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{N_{\text{rollout}}} \sum_t \frac{1}{|G|} \sum_{v \in G} \sum_{i=1}^{N_{\text{var}}} \left[\log \sigma^t_{v,i} + \frac{\left(X^t_{v,i} - \mu^t_{v,i} \right)^2}{2 \left(\sigma^t_{v,i} \right)^2} \right] \tag{20}$$

Training uses AdamW (learning rate 0.001), batch size 8 (limited area), 500 epochs for single-step and 200 for rollout fine-tuning, requiring 3–4 days on an NVIDIA A100 GPU (80 GB). All variables are normalised to zero mean and unit variance using training statistics.

H.3. Inference and autoregressive forecasting

Given initial states $X^{-1:0}$ and forcing inputs $F^{1:T}$, models autoregressively predict $X^{t+1} = \hat{f}(X^{t-1:t}, F^{t+1})$. Limited area models use lateral boundary forcing: predictions within a 10-grid-cell boundary are replaced with ground truth at each step. Forcing inputs include solar radiation at the top-of-atmosphere, diurnal and annual cycle encodings (sine/cosine transforms), and open water fraction. A 57 hour limited area forecast requires only 1.5 s on a single A100 GPU.

H.4. CP application

H.4.1. Nonconformity scores

For Graph-FM (MSE) (AER):

$$s(X,Y) = |Y - \hat{f}(X)|, \quad \mathbb{C}^{\alpha}(X) = \left[\hat{f}(X) - \hat{q}, \hat{f}(X) + \hat{q}\right] \tag{21}$$

For Graph-FM (NLL) (STD):

$$s(X,Y) = \frac{|Y - \mu(X)|}{\sigma(X)}, \quad \mathbb{C}^{\alpha}(X) = [\mu(X) - \hat{q} \cdot \sigma(X), \mu(X) + \hat{q} \cdot \sigma(X)] \tag{22}$$

This calibrates the model's potentially miscalibrated uncertainty estimates.

H.4.2. Calibration procedure

CP is performed independently for each spatio-temporal cell. Limited area models calibrate $19 \times 238 \times 268 \times 17 = 20{,}602{,}232$ cells using 270 September 2021 forecasts; global models use 730 forecasts from 2018. For each cell, the $(1-\alpha)$-quantile is computed:

$$\hat{q} = F_{\hat{s}}^{-1}\left(\frac{\lceil (n+1)(1-\alpha) \rceil}{n}\right). \tag{23}$$

Calibration is computationally efficient: 229–310 s for limited area models and 366–401 s for global models, including score computation, quantile calculation, and coverage validation.

H.5. Exchangeability considerations

H.5.1. Weather forecasting as an initial boundary value problem

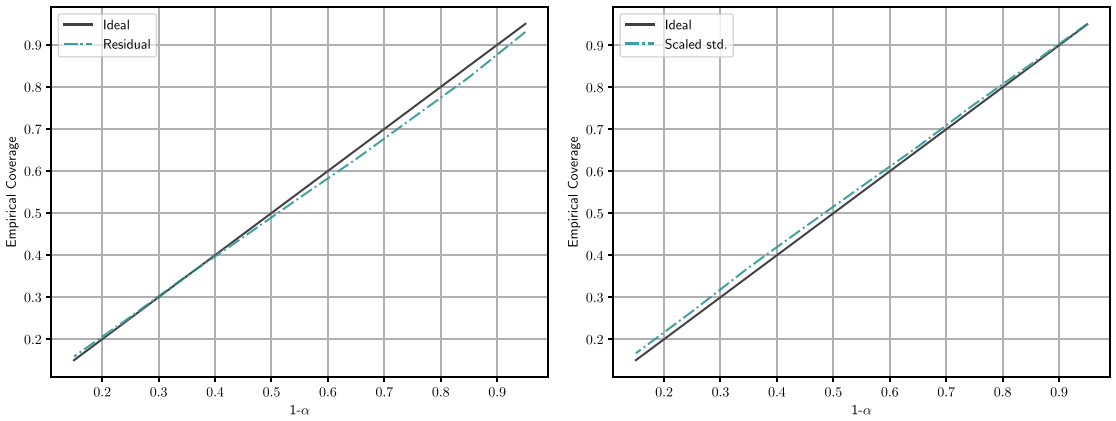
Applying CP to time series typically violates exchangeability due to temporal dependencies. However, treating each forecast as an independent initial boundary value problem (IBVP) preserves exchangeability: each forecast is fully determined by its initial conditions $X^{-1:0}$ and forcing $F^{1:T}$, making forecasts initialised at sufficiently separated times independent realisations from the atmospheric state space. This holds even when forecasts cover overlapping time periods, since each prediction conditions only on its own initial state.

CP is applied to the entire spatio-temporal output tensor $Y \in \mathbb{R}^{T_{\text{out}} \times N_x \times N_y \times N_{\text{var}}}$ simultaneously, preserving the physical forecast structure.

H.5.2. Calibration-to-test exchangeability

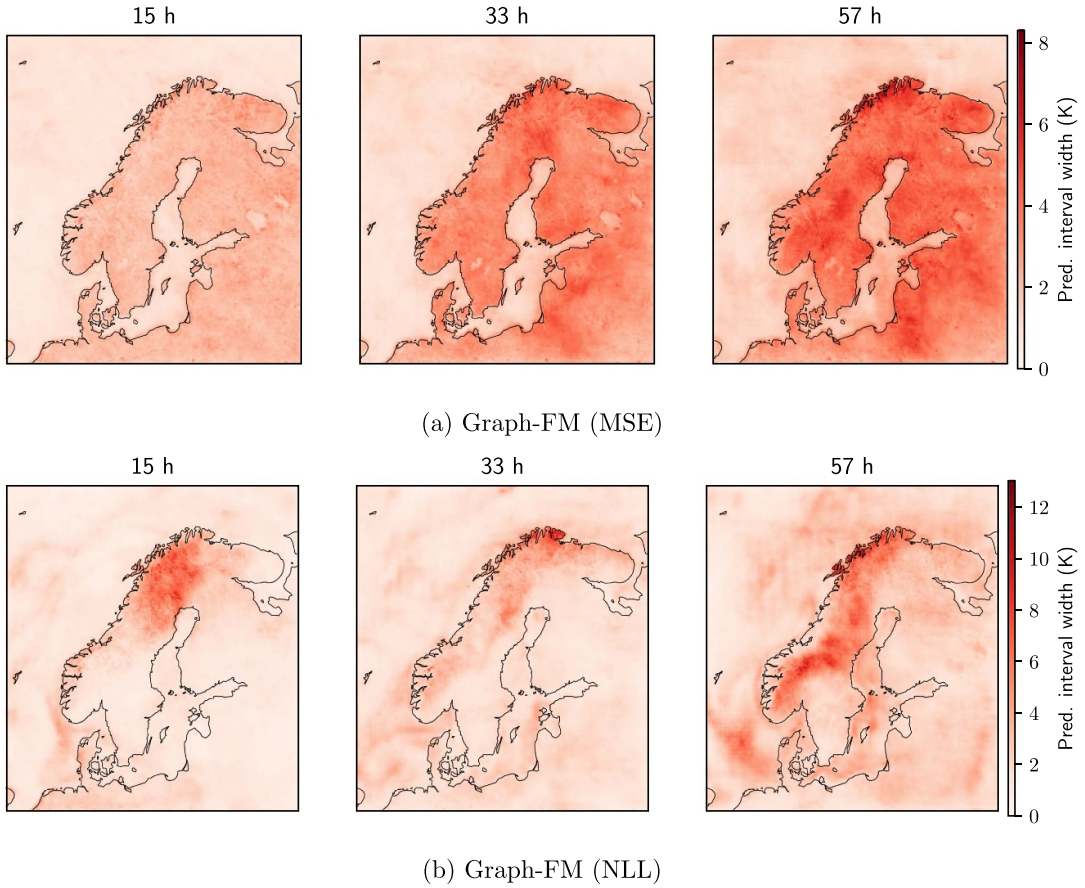
The coverage guarantee requires test forecasts to be exchangeable with calibration forecasts. For limited area models, this assumes September 2022 patterns are exchangeable with September 2021 patterns; for global models, 2019 states are exchangeable with 2018 states. These assumptions are reasonable given year-to-year climate consistency, validated by the excellent empirical coverage observed. Potential violations include anomalous events not represented in calibration data, systematic climate shifts, or changes in model distribution, warranting periodic recalibration with recent data.

To investigate the setting where there is a distributional shift between the calibration and test datasets, we consider a setting where we shift the month of the year in the weather data. For this experiment, we calibrate the limited area weather model using predictions for September 2021, and evaluate on January 2023. The weather in the Nordic region being modelled exhibits strong seasonal dependencies, with the winter month of January being much colder than the September month used for calibration. Note that while we only use September for calibration, the model has been trained with data from all seasons of the year. Empirical coverage is shown in figure 33, and in figure 34 we plot the width of



(a) Graph-FM (MSE) (b) Graph-FM (NLL)

Figure 33. Empirical coverage for limited area weather forecasting models, when calibrated on September and evaluating coverage on January.



(a) Graph-FM (MSE)

(b) Graph-FM (NLL)

Figure 34. Width of predictive intervals in an example January forecast from the distributional shift setting. Here $\alpha = 0.05$ and we show results for temperature 2 m above ground (`2t`).

predictive intervals for 2 m temperature predictions. We note that, while the coverage shows larger deviations than in the setting where both calibration and evaluation were performed for the September data, the method overall still provides useful uncertainty estimates. This may be attributed to the reason that the physics regime characterising the weather evolution does not change across the months, but only the initial conditions to which the model is exposed.

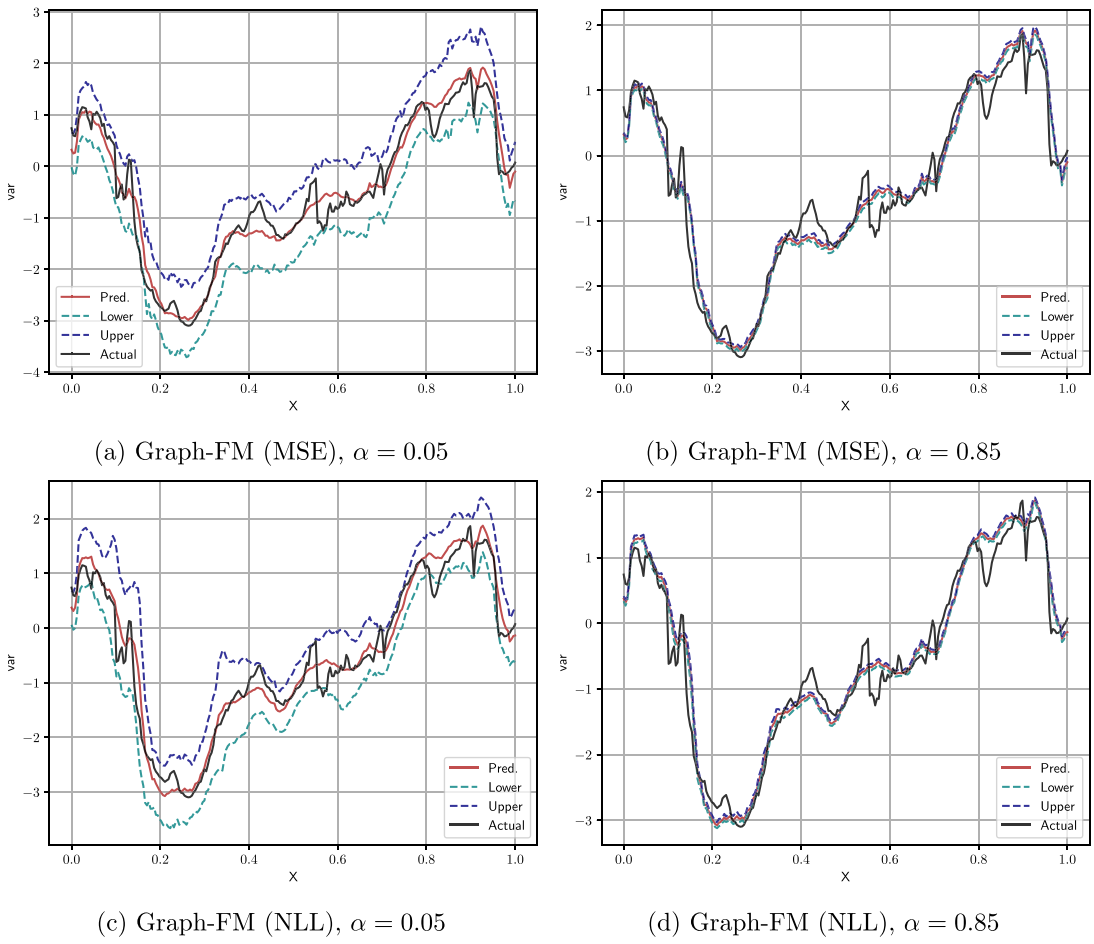


Figure 35. Slice plots across the *x*-axis showing u-wind component predictions. Panels (a)–(d) show ground truth, prediction, and CP-derived upper/lower bounds for Graph-FM (MSE) and Graph-FM (NLL) at 95% ($\alpha = 0.05$) and 15% ($\alpha = 0.85$) coverage.

H.6. Additional results

Figure 35 shows spatial slices illustrating error bars at different coverage levels. The Graph-FM (NLL) model produces tighter, spatially variable error bars due to input-dependent uncertainties, while Graph-FM (MSE) produces constant-width bars at each lead time. Both achieve target coverage, validating the CP framework.

Figure 36 demonstrates empirical coverage for global forecasting (temperature and geopotential at 700 hPa). Both model types closely follow the ideal diagonal, confirming guaranteed marginal coverage across 1,161,600 spatial–temporal points per variable.

H.6.1. Deterministic vs probabilistic models

Deterministic models (AER): Error bars have constant width at each lead time, determined purely by calibration performance. Computationally efficient but potentially overconservative in predictable regions.

Probabilistic models (STD): Error bars adapt to local forecast uncertainty, providing tighter bounds in high-confidence regions. Requires NLL training but benefits from CP calibration of potentially miscalibrated model uncertainties.

Both approaches achieve guaranteed coverage, with probabilistic models generally providing more informative bounds at the cost of additional complexity.

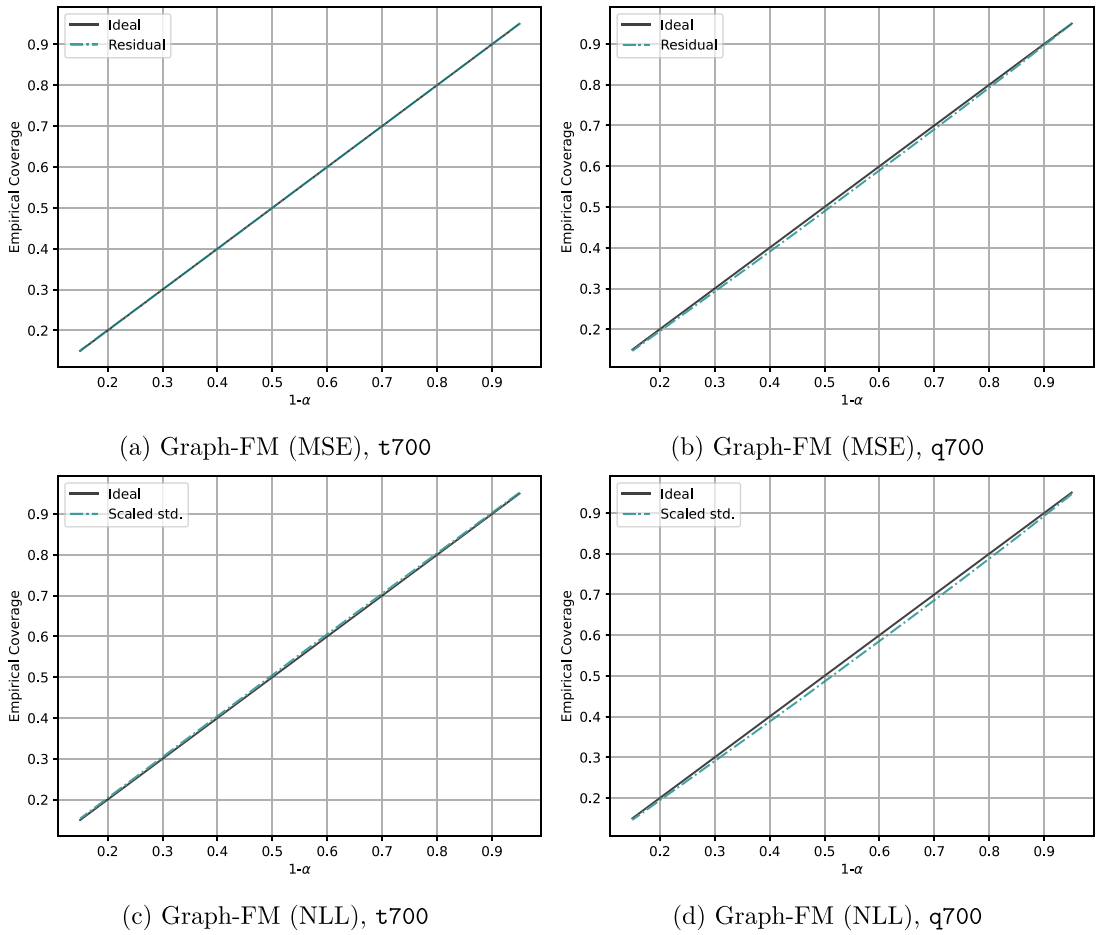


Figure 36. Empirical coverage for global weather forecasting models showing temperature (t700) and geopotential (q600) at 700 hPa. Both models achieve near-ideal coverage across all α levels.

Appendix I. Camera diagnostic on a tokamak

I.1. Camera configuration and data processing

I.1.1. Camera specifications

The MAST was equipped with Photron fast-visible cameras capturing plasma evolution at an average temporal resolution of 1.2 ms. We focus on the RBB camera that prioritises a central solenoid view of the plasma. It provides a panoramic view of the reactor, showing the poloidal cross-sectional layout on both sides of the central solenoid with spatial resolution of 448×640 pixels.

This camera is tuned to visible wavelengths and primarily captures Balmer D_α light emitted from the plasma edge, enabling visualisation of complex MHD dynamics, including edge-localised modes (ELMs).

I.1.2. Data selection criteria

Shot selection for training and testing followed stringent criteria to ensure data quality and consistency:

1. **Plasma presence:** Only shots containing actual confined plasma were selected, excluding commissioning pulses and dummy shots.
2. **Temporal duration:** Shots must contain more than 100 time steps to capture sufficient plasma evolution dynamics.
3. **Camera calibration consistency:** All selected shots must share identical camera calibration parameters. Using CalCam (Silburn *et al* 2022) combined with 3D CAD models of MAST, the domain range covered by each camera calibration was determined. The upper and lower limits of the domain range were mapped onto uniform grids in both R and Z axes.
4. **Spatial resolution uniformity:** All camera images must have identical spatial resolution to maintain consistency in the training dataset.

Fifty-five shots from the range 30 250–30 431 in the M9 campaign were selected (50 for training, 5 for testing). The diversity of plasma scenarios in the training data is illustrated in figure 19 in Gopakumar *et al* (2024), which shows the temporal evolution of plasma current and total heating power across the selected shots.

I.2. Training regime: sequential time window approach

I.2.1. Rationale for non-autoregressive architecture

Unlike Markovian simulations, camera data presents fundamentally different characteristics that necessitate an alternative training approach:

- **Non-Markovian dynamics:** The plasma evolution captured by cameras represents a non-Markovian process with inherent noise and partial system information.
- **Absence of causal information:** The dataset does not include explicit information about control inputs (coil currents, heating parameters) that influence plasma evolution.
- **Continuous data generation:** Unlike simulations, where initial conditions fully determine evolution, experimental data is continuously generated during the plasma shot, providing new information at each time step.
- **Error accumulation:** Autoregressive rollout would lead to rapid error accumulation over longer time horizons, limiting prediction capability.

I.2.2. Sequential window mapping

Instead of autoregressive prediction, we employ a sliding time window approach. The FNO maps a fixed-length input sequence to a fixed-length output sequence:

$$\tilde{Y}_{t:t+\text{step}} = \mathcal{F}_\theta\left(X_{t-T_{\text{in}}:t}, \mathcal{G}\right) \tag{24}$$

where:

- $X_{t-T_{\text{in}}:t} \in \mathbb{R}^{T_{\text{in}} \times N_x \times N_y}$ represents the input sequence of camera frames
- $\tilde{Y}_{t:t+\text{step}} \in \mathbb{R}^{\text{step} \times N_x \times N_y}$ represents the predicted future frames
- $\mathcal{G} \in \mathbb{R}^{N_x \times N_y \times 2}$ represents the spatial grid discretisation
- $\mathcal{F}_\theta$ denotes the FNO with parameters θ

For our experiments, we set $T_{\text{in}} = 10$ and step $= 10$, allowing the FNO to predict 12 ms of plasma evolution given the previous 12 ms of camera data.

I.2.3. Training data construction

Each plasma shot containing approximately 200 frames is converted into multiple overlapping input–output pairs through a sliding window mechanism with a stride of 1:

$$\text{Pair}_1 : X_{[1:10]} \rightarrow Y_{[11:20]} \tag{25}$$

$$\text{Pair}_2 : X_{[2:11]} \rightarrow Y_{[12:21]} \tag{26}$$

$$\vdots \tag{27}$$

$$\text{Pair}_k : X_{[k:k+9]} \rightarrow Y_{[k+10:k+19]}. \tag{28}$$

This strategy yields approximately 180 training pairs per shot, significantly increasing the effective dataset size. The FNO learns to predict plasma evolution from any intermediate state within the operational range, rather than solely from initial conditions.

I.3. Real-time deployment strategy

I.3.1. Inference pipeline

For real-time operation, the trained FNO is deployed with the following pipeline:

1. **Initial phase:** Collect the first $T_{\text{in}} = 10$ camera frames at plasma startup
2. **Prediction:** Generate predictions for the next step $= 10$ frames using the FNO
3. **Update:** As new frames become available from the camera, slide the input window forward by one frame
4. **Iteration:** Repeat prediction with the updated window throughout the plasma shot duration

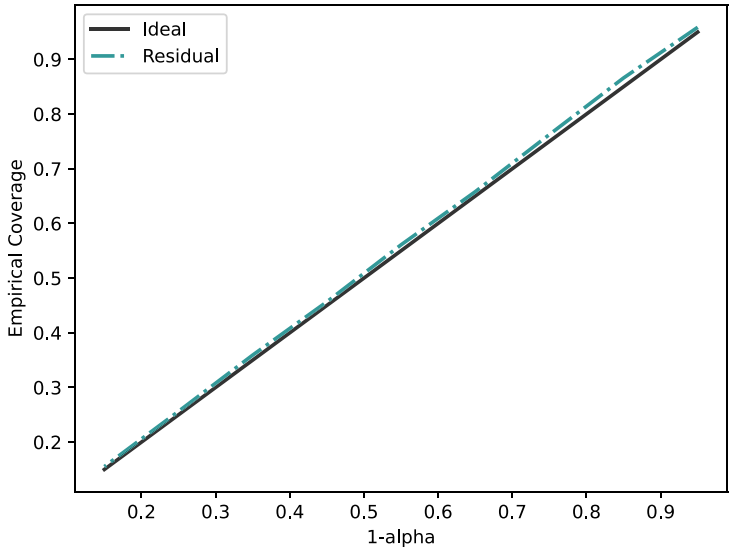


Figure 37. Empirical coverage guaranteed by applying CP over the FNO forecasting the plasma evolution.

Since camera data is acquired at 1.2 ms intervals and the FNO requires only 6 ms for inference, predictions can be generated faster than real-time, providing a 12 ms forecast horizon with 6 ms computational overhead.

I.4. Exchangeability considerations for time series data

I.4.1. Theoretical framework

Traditional CP requires exchangeability between calibration and test data (Vovk *et al* 2005). For time-series data, this assumption is typically violated. However, by treating each plasma shot as an initial boundary value problem (IBVP), we maintain exchangeability through the following reasoning:

- **IBVP structure:** Each input–output pair (X_i, Y_i) represents a self-contained prediction task where the output is fully determined by the input sequence and boundary conditions, if they are there (forcing terms from control systems).
- **Temporal independence:** Plasma profile predictions starting at different times (e.g. 30 s into the shot or 60 s into the shot) are independent, conditioned on their respective initial states.
- **Distributional sampling:** The calibration dataset represents samples from a large distribution characterising the entire operational range of the tokamak under consideration.

I.4.2. Limitations and assumptions

The exchangeability assumption relies on several critical conditions:

$$\mathbb{P}\left(\text{Exchangeable}\right) = \begin{cases} \text{High} & \text{if shots share similar plasma profiles} \\ \text{Low} & \text{if shots have dissimilar characteristics} \end{cases} \tag{29}$$

As demonstrated in figure 21, when the prediction shot has significantly different plasma discharge profiles from the calibration shots, exchangeability is violated and coverage degrades. This necessitates calibration datasets that adequately span the operational space of interest.

References

Abdar M *et al* 2021 A review of uncertainty quantification in deep learning: techniques, applications and challenges *Inf. Fusion* **76** 243–97

Alhajeri M S, Abdullah F, Wu Z and Christofides P D 2022 Physics-informed machine learning modeling for predictive control using noisy data *Chem. Eng. Res. Des.* **186** 34–49

Alkin B, Fürst A, Schmid S, Gruber L, Holzleitner M and Brandstetter J 2024 Universal physics transformers: a framework for efficiently scaling neural operators (arXiv:2402.12365)

Angelopoulos A N and Bates S 2023 Conformal prediction: a gentle introduction *Found. Trends Mach. Learn.* **16** 494–591

Azizzadenesheli K, Kovachki N, Li Z, Liu-Schiaffini M, Kossaifi J and Anandkumar A 2024 Neural operators for accelerating scientific simulations and design *Nat. Rev. Phys.* **6** 320–8

Balch M S 2012 Mathematical foundations for a theory of confidence structures *Int. J. Approx. Reason.* **53** 1003–19

Baldi P, Cranmer K, Faucett T, Sadowski P and Whiteson D 2016 Parameterized neural networks for high-energy physics *Eur. Phys. J. C* **76** 235
Begoli E, Bhattacharya T and Kusnezov D 2019 The need for uncertainty quantification in machine-assisted medical decision making *Nat. Mach. Intell.* **1** 20–23
Bellan P M 2006 *Fundamentals of Plasma Physics* (Cambridge University Press)
Bertone G, Deisenroth M P, Kim J S, Liem S, Ruiz de Austri R and Welling M 2019 Accelerating the bsm interpretation of lhc data with machine learning *Phys. Dark Univ.* **24** 100293
Bi K, Xie L, Zhang H, Chen X, Gu X and Tian Q 2023 Accurate medium-range global weather forecasting with 3D neural networks *Nature* **619** 1–6
Bommasani R *et al* 2022 On the opportunities and risks of foundation models (arXiv:2108.07258)
Bouallègue Z B *et al* 2024 The rise of data-driven weather forecasting: a first statistical assessment of machine learning-based weather forecasts in an operational-like context *Bull. Am. Meteorol. Soc.* **105** E864–83
Carey N, Zanisi L, Pamela S, Gopakumar V, Omotani J, Buchanan J and Brandstetter J 2024 Data efficiency and long term prediction capabilities for neural operator surrogate models of core and edge plasma codes (arXiv:2402.08561)
Cella L and Martin R 2022 Validity, consonant plausibility measures and conformal prediction *Int. J. Approx. Reason.* **141** 110–30
Chandrasekhar S 1943 Stochastic problems in physics and astronomy *Rev. Mod. Phys.* **15** 1–89
Chen K *et al* 2023a FengWu: pushing the skillful global medium-range weather forecast beyond 10 days lead (arXiv:2304.02948)
Chen L, Zhong X, Zhang F, Cheng Y, Xu Y, Qi Y and Li H 2023b Fuxi: a cascade machine learning forecasting system for 15-day global weather forecast *npj Clim. Atmos. Sci.* **6** 190
Coiffier J 2011 *Fundamentals of Numerical Weather Prediction* (Cambridge University Press)
Danabasoglu G *et al* 2020 The community earth system model version 2 (cesm2) *J. Adv. Modeling Earth Syst.* **12** e2019MS001916
Degrave J *et al* 2022 Magnetic control of tokamak plasmas through deep reinforcement learning *Nature* **602** 414–9
Diquigiovanni J, Fontana M and Vantini S 2021 Conformal prediction bands for multivariate functional data *J. Multivar. Anal.* **189** 104879
Ebi K L *et al* 2021 Extreme weather and climate change: population health and health system implications *Annu. Rev. Public Health* **42** 293–315
Gal Y and Ghahramani Z 2016 Dropout as a bayesian approximation: representing model uncertainty in deep learning *Int. Conf. on Machine Learning*
Geneva N and Zabaras N 2020 Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks *J. Comput. Phys.* **403** 109056
Giudicelli G *et al* 2024 3.0 - MOOSE: enabling massively parallel multiphysics simulations *SoftwareX* **26** 101690
Gopakumar V *et al* (the JOREK Team and M. Team) 2024 Plasma surrogate modelling using Fourier neural operators *Nucl. Fusion* **64** 056025
Gopakumar V, Gray A, Zanisi L, Nunn T, Giles D, Kusner M, Pamela S and Deisenroth M P 2025 Calibrated physics-informed uncertainty quantification *42nd Int. Conf. on Machine Learning* (available at: https://openreview.net/forum?id = Z2uLBBck2X)
Gopakumar V, Pamela S and Samaddar D 2023 Loss landscape engineering via data regulation on pinns *Mach. Learn. Appl.* **12** 100464
Gopakumar V and Samaddar D 2020 Image mapping the temporal evolution of edge characteristics in tokamaks using neural networks *Mach. Learn.: Sci. Technol.* **1** 015006
Graubner A, Azizzadenesheli K K, Pathak J, Mardani M, Pritchard M, Kashinath K and Anandkumar A 2022 Calibration of large neural weather models *NeurIPS 2022 Workshop on Tackling Climate Change With Machine Learning*
Gupta J K and Brandstetter J 2023 Towards multi-spatiotemporal-scale generalized PDE modeling *Trans. Mach. Learn. Res.* (https://doi.org/10.48550/arXiv.2209.15616)
Hackbusch W 2017 *The Poisson Equation* (Springer) pp 29–42
Ham C, Kirk A and Veruagh K 2022 Insights on disruption physics in mast using high speed visible camera data *IAEA Second Technical Meeting on Plasma Disruptions and Their Mitigation* (available at: https://conferences.iaea.org/event/281/contributions/24423/)
Hao Z, Su C, Liu S, Berner J, Ying C, Su H, Anandkumar A, Song J and Zhu J 2024 Dpot: auto-regressive denoising operator transformer for large-scale pde pre-training *Int. Conf. on Machine Learning* (*Vienna, 2024*)
Haykin S 1994 *Neural Networks: A Comprehensive Foundation* (Prentice Hall PTR)
Hersbach H *et al* 2020 The ERA5 global reanalysis *Q. J. R. Meteorol. Soc.* **146** 1999–2049
Hoelzl M *et al* 2021 The jorek non-linear extended mhd code and applications to large-scale instabilities and their control in magnetically confined fusion plasmas *Nucl. Fusion* **61** 065001
Hose D and Hanss M 2021 A universal approach to imprecise probabilities in possibility theory *Int. J. Approx. Reason.* **133** 133–58
Hospital A, Goni J R, Orozco M and Gelpí J L 2015 Molecular dynamics simulations: advances and applications *Adv. Appl. Bioinform. Chem.* **8** 37–47
Howlett L, Cziegler I, Freethy S and Meyer H (the MAST team) 2023 L-h transition studies on mast: power threshold and heat flux analysis *Nucl. Fusion* **63** 052001
Hu Y, Chen L, Wang Z and Li H 2023 Swinvrnn: a data-driven ensemble forecasting model via learned distribution perturbation *J. Adv. Modeling Earth Syst.* **15** e2022MS003211
Jiang C M, Esmaeilzadeh S, Azizzadenesheli K, Kashinath K, Mustafa M, Tchelepi H A, Prabhat P M and Anandkumar A 2020 Meshfreeflownet: a physics-constrained deep continuous space-time super-resolution framework (arXiv:2005.01463)
Johansson U, Boström H and Löfström T 2021 Investigating normalized conformal regressors *2021 IEEE Symp. Series on Computational Intelligence (SSCI)* pp 01–08
Kalnay E 2002 *Atmospheric Modeling, Data Assimilation and Predictability* (Cambridge University Press)
Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nat. Rev. Phys.* **3** 422–40
Kates-Harbeck J, Svyatkovskiy A and Tang W 2019 Predicting disruptive instabilities in controlled fusion plasmas through deep learning *Nature* **568** 526–31
Keisler R 2022 Forecasting global weather with graph neural networks (arXiv:2202.07575)
Kingma D P and Ba J 2015 Adam: a method for stochastic optimization *3rd Int. Conf. on Learning Representations, ICLR2015, (San Diego, CA, USA, 7 May–9 May 2015)* (*Conf. Track Proc.*) ed Y Bengio and Y LeCun (available at: http://arxiv.org/abs/1412.6980)
Kirk A *et al* (the MAST team) 2006 Filament structures at the plasma edge on mast *Plasma Phys. Control. Fusion* **48** B433
Knight J 2002 Safety critical systems: challenges and directions *Proc. 24th Int. Conf. on Software Engineering. ICSE 2002* pp 547–50
Koenker R 2005 *Quantile Regression* (*Econometric Society Monographs*) (Cambridge University Press)

Kurth T, Subramanian S, Harrington P, Pathak J, Mardani M, Hall D, Miele A, Kashinath K and Anandkumar A 2023 Fourcastnet: accelerating global high-resolution weather forecasting using adaptive fourier neural operators *Proc. Platform for Advanced Scientific Computing Conf., PASC '23, (New York, NY, USA)* (Association for Computing Machinery)

Lakshminarayanan B, Pritzel A and Blundell C 2017 Simple and scalable predictive uncertainty estimation using deep ensembles *Neural Information Processing Systems* (*Long Beach, CA, USA, 2017*) (https://doi.org/10.48550/arXiv.1612.01474)

Lalonde E R, Vischschraper B, Bitsuamlak G and Dai K 2021 Comparison of neural network types and architectures for generating a surrogate aerodynamic wind turbine blade model *J. Wind Eng. Ind. Aerodyn.* **216** 104696

Lam R *et al* 2023 Learning skillful medium-range global weather forecasting *Science* **382** 1416–21

Lavin A *et al* 2021 Simulation intelligence: towards a new generation of scientific methods (arXiv:2112.03235)

Lei J, G'Sell M, Rinaldo A, Tibshirani R J and Wasserman L 2018 Distribution-free predictive inference for regression *J. Am. Stat. Assoc.* **113** 1094–111

Lerede D, Nicoli M, Savoldi L and Trotta A 2023 Analysis of the possible contribution of different nuclear fusion technologies to the global energy transition *Energy Strat. Rev.* **49** 101144

Li Z *et al* 2023 Geometry-informed neural operator for large-scale 3D PDEs *37th Conf. on Neural Information Processing Systems* (available at: https://openreview.net/forum?id = 86dXbqT5Ua)

Li Z, Kovachki N B, Azizzadenesheli K, liu B, Bhattacharya K, Stuart A and Anandkumar A 2021 Fourier neural operator for parametric partial differential equations *Int. Conf. on Learning Representations* (available at: https://openreview.net/forum?id = c8P9NQVtmnO)

Linke J, Du J, Loewenhoff T, Pintsuk G, Spilker B, Steudel I and Wirtz M 2019 Challenges for plasma-facing components in nuclear fusion *Matter Radiat. Extremes* **4** 056201

Ma Z, Azizzadenesheli K and Anandkumar A 2024 Calibrated uncertainty quantification for operator learning via conformal prediction (arXiv:2402.01960)

MacKay D J C 1992 A practical Bayesian framework for backpropagation networks *Neural Comput.* **4** 448–72

Mánek P, Goffrier G V, Gopakumar V, Nikolaou N, Shimwell J and Waldmann I 2023 Fast regression of the tritium breeding ratio in fusion reactors *Mach. Learn.: Sci. Technol.* **4** 015008

Martin R 2019 False confidence, non-additive beliefs and valid statistical inference *Int. J. Approx. Reason.* **113** 39–73

McCabe M *et al* 2023 Multiple physics pretraining for physical surrogate models *NeurIPS 2023 AI for Science Workshop* (available at: https://openreview.net/forum?id = M12lmQKuxa)

Messoudi S, Destercke S and Rousseau S 2021 Copula-based conformal prediction for multi-target regression *Pattern Recognit.* **120** 108101

Messoudi S, Destercke S and Rousseau S 2022 Ellipsoidal conformal inference for multi-target regression *Proc. 11th Symp. on Conformal and Probabilistic Prediction With Applications* (*Proc. of Machine Learning Research* vol 179) ed U Johansson, H Boström, K An Nguyen, Z Luo and L Carlsson (PMLR) (available at: https://proceedings.mlr.press/v179/messoudi22a.html) pp 294–306

Müller M *et al* 2017 AROME-MetCoOp: a nordic convective-scale operational weather prediction model *Weather Forecast.* **32** 609–27

Oskarsson J, Landelius T, Deisenroth M P and Lindsten F 2024 Probabilistic weather forecasting with hierarchical graph neural networks (arXiv:2406.04759)

Pamela S J P *et al* (the JOREK Team) 2024 Neural-parareal: dynamically training neural operators as coarse solvers for time-parallelisation of fusion mhd simulations (arXiv:2405.01355)

Papadopoulos H 2008 Inductive conformal prediction: theory and application to neural networks *Tools in Artificial Intelligence* ed, editor P Fritzsche (IntechOpen) ch 18

Pfaff T, Fortunato M, Sanchez-Gonzalez A and Battaglia P 2021 Learning mesh-based simulation with graph networks *Int. Conf. on Learning Representations* (available at: https://openreview.net/forum?id = roNqYL0_XP)

Price I, Sanchez-Gonzalez A, Alet F, Ewalds T, El-Kadi A, Stott J, Mohamed S, Battaglia P, Lam R and Willson M 2023 Gencast: diffusion-based ensemble forecasting for medium-range weather (arXiv:2312.15796)

Psaros A F, Meng X, Zou Z, Guo L and Karniadakis G E 2023 Uncertainty quantification in scientific machine learning: Methods, metrics and comparisons *J. Comput. Phys.* **477** 111902

Quionero-Candela J, Sugiyama M, Schwaighofer A and Lawrence N D 2009 *Dataset Shift in Machine Learning* (The MIT Press)

Rahman M A *et al* 2024 Pretraining codomain attention neural operators for solving multiphysics pdes *Neural Information Processing Systems* (*Vancouver, Canada, 2024*) (https://doi.org/10.48550/arXiv.2403.12553)

Rasp S *et al* 2024 Weatherbench 2: a benchmark for the next generation of data-driven global weather models (arXiv:2308.15560)

Romano Y, Patterson E and Candes E 2019 Conformalized quantile regression *Advances in Neural Information Processing Systems* vol 32, ed H Wallach, H Larochelle, A Beygelzimer, F d' Alchè-Buc, E Fox and R Garnett (Curran Associates, Inc.) (available at: https://proceedings.neurips.cc/paper_files/paper/2019/file/5103c3584b063c431bd1268e9b5e76fb-paper.pdf)

Ronneberger O, Fischer P and Brox T 2015 U-net: Convolutional networks for biomedical image segmentation *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, ed N Navab, J Hornegger, W M Wells and A F Frangi (Springer) pp 234–41

Scarselli F, Gori M, Tsoi A C, Hagenbuchner M and Monfardini G 2009 The graph neural network model *IEEE Trans. Neural Netw.* **20** 61–80

Shafer G and Vovk V 2008 A tutorial on conformal prediction *J. Mach. Learn. Res.* **9** 371–421

Sheshadri A, Borrus M, Yoder M and Robinson T 2021 Midlatitude error growth in atmospheric GCMS: the role of eddy growth rate *Geophys. Res. Lett.* **48** e2021GL096126

Shukla K, Oommen V, Peyvan A, Penwarden M, Plewacki N, Bravo L, Ghoshal A, Kirby R M and Karniadakis G E 2024 Deep neural operators as accurate surrogates for shape optimization *Eng. Appl. Artif. Intell.* **129** 107615

Silburn S *et al* 2022 *Zenodo* (available at: https://doi.org/10.5281/zenodo.6891504)

Stankeviciute K, Alaa A M and van der Schaar M 2021 Conformal time-series forecasting *Advances in Neural Information Processing Systems* vol 34, ed M Ranzato, A Beygelzimer, Y Dauphin, P Liang and J W Vaughan (Curran Associates, Inc.) (available at: https://proceedings.neurips.cc/paper_files/paper/2021/file/312f1ba2a72318edaaa995a67835fad5-paper.pdf) pp 6216–28

Sun S 2022 Conformal methods for quantifying uncertainty in spatiotemporal data: a survey (arXiv:2209.03580)

Tibshirani R J, Barber R F, Candes E and Ramdas A 2019 Conformal prediction under covariate shift *Advances in Neural Information Processing Systems*

Tipler P A 2008 *Physics for Scientists and Engineers. Volume 1, Mechanics, Oscillations and Waves, Thermodynamics* 6th edn (W.H. Freeman)

van de Plassche K L et al 2020 Fast modeling of turbulent transport in fusion plasmas using neural networks *Phys. Plasmas* **27** 022310
Vovk V 2012 Conditional validity of inductive conformal predictors *Asian Conf. on Machine Learning*
Vovk V, Gammerman A and Shafer G 2005 *Algorithmic Learning in a Random World* (Springer)
Walkden N, Riva F, Harrison J, Militello F, Farley T, Omotani J and Lipschultz B 2022 The physics of turbulence localised to the tokamak divertor volume *Commun. Phys.* **5** 139
Wen G, Li Z, Long Q, Azizzadenesheli K, Anandkumar A and Benson S M 2023 Real-time high-resolution CO_2 geological storage prediction using nested fourier neural operators *Energy Environ. Sci.* **16** 1732–41
Xu C and Xie Y 2021 Conformal prediction interval for dynamic time-series *Proc. 38th Int. Conf. on Machine Learning* (*Proc. Machine Learning Research* vol 139), ed M Meila and T Zhang (PMLR) (available at: https://proceedings.mlr.press/v139/xu21h.html) pp 11559–69
Xu C, Xie Y, Vazquez D A Z, Yao R and Qiu F 2023 Spatio-temporal wildfire prediction using multi-modal data *IEEE J. Sel. Areas Inf. Theory* **4** 302–13
Yin H, Vahdat A, Alvarez J M, Mallya A, Kautz J and Molchanov P 2022b A-vit: adaptive tokens for efficient vision transformer *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 10809–18
Yin H, Vahdat A, Alvarez J, Mallya A, Kautz J and Molchanov P 2022a A-ViT: adaptive tokens for efficient vision transformer *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*
Yin Z, Orozco R, Louboutin M and Herrmann F J 2023 Solving multiphysics-based inverse problems with learned surrogates and constraints *Adv. Modeling Simul. Eng. Sci.* **10** 14
Zou Z, Meng X, Psaros A F and Karniadakis G E 2024 Neuraluq: a comprehensive library for uncertainty quantification in neural differential equations and operators *SIAM Rev.* **66** 161–90
Zwicker D 2020 py-pde: a python package for solving partial differential equations *J. Open Source Softw.* **5** 2158