Guaranteed prediction sets for functional surrogate models

Ander Gray[1] **Vignesh Gopakumar**[2,3] **Sylvain Rousseau**[1] **Sébastien Destercke**[1]

[1]Université de technologie de Compiègne, CNRS, Heudiasyc, France
[2]UK Atomic Energy Authority
[3]University College London, UK

Abstract

We propose a method for obtaining statistically guaranteed prediction sets for functional machine learning methods: surrogate models which map between function spaces, motivated by the need to build reliable PDE emulators. The method constructs nested prediction sets on a low-dimensional representation (an SVD) of the surrogate model's error, and then maps these sets to the prediction space using set-propagation techniques. The result is conformal prediction coverage guaranteed prediction sets for functional surrogate models. We use zonotopes as basis of the set construction, which allow an exact linear propagation and are closed under Cartesian products, making them well-suited to this high-dimensional problem. The method is model agnostic and can thus be applied to complex Sci-ML models, including Neural Operators, but also in simpler settings. We also introduce a technique to capture the truncation error of the SVD, preserving the guarantees of the method.

1 INTRODUCTION

One struggles to find an engineering or scientific discipline which has remained untouched from the rapid progression in machine learning (ML) and artificial intelligence (AI). Scientific machine learning (Sci-ML), with tailored architectures for physics-based problems, have in particular driven major advancements. Neural Operators (NOs) [Li et al., 2020], neural networks which can learn between function spaces, have received attention due to their efficient surrogacy of partial differential equation (PDE) solvers [Azizzadenesheli et al., 2024], and have seen application in complex problems including weather modelling [Kurth et al., 2023] and plasma physics [Gopakumar et al., 2024b].

However, the wide application of AI methods continues despite widely-shared concerns, as adumbrated by Brundage et al. [2020], Dalrymple et al. [2024], and many others, about the reliability and soundness of these methods. This is an opinion that we share: that methods for AI reliability are underdeveloped in comparison to its progression and wide application.

Although multiple methods exist for uncertainty analysis in AI, little R&D effort has gone into approaches which can provide *quantitative safety guarantees* on the predictions of AI systems. Probabilistic machine learning methods, such as Bayesian neural networks, Gaussian processes, Monte Carlo drop-out, and deep ensembles, are powerful methods which can equip predictions with distributional uncertainties. These include many notable works in neural PDE surrogates and operator learning, including [Yang et al., 2022, Yang and Perdikaris, 2019, Beltran et al., 2024]

However, they do not attempt to provide statistical guarantees. By *statistical guarantee* we refer to a provable property of the model's uncertainty, given some quite weak assumptions about the randomness of the data. In this paper, we pursue a method based on conformal prediction [Vovk et al., 2005, Shafer and Vovk, 2008], which is less committal than purely probabilistic approaches, but can yield such guarantees. Rather than producing a full predictive distribution, the method gives a *set-valued* prediction $\mathbb{C}^\alpha$ equipped with a confidence level $1-\alpha$. This prediction set can be *guaranteed* to contain the next unobserved true label Y_{n+1}, with at least probability $1-\alpha$:

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

The prediction set $\mathbb{C}^\alpha$ is constructed using previously observed data (X_i, Y_i), and the guarantee holds if the sequence $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, Y_{n+1})$ is exchangeable, e.g., if the new data point is drawn from the same joint distribution.

This paper's goal is to extend this framework to models with function-valued outputs. That is, given an input X, we want to predict not a single scalar, but a full function F

(large vector or tensor e.g., a time series, a spatial field, or a parameterized curve). Our goal is to construct a set of functions that is guaranteed, with a user-defined probability, to contain the true function associated with a new input. Our primary motivation for this is to develop reliable uncertainty quantification for PDE surrogate models, where the functional data represents solutions from numerical PDE solvers. However, the method could be faithfully applied to non-physics cases. Unlike scalar predictions, functional data can exhibit complex dependencies across their domain (e.g., neighboring time points or spatial locations), which our method must capture to produce meaningful predictions. Finally, we want this guarantee to hold using only a held-out calibration dataset. The following formal setup further describes this problem.

Problem statement Given a pre-trained model $\hat{f} : X \mapsto F$, which maps to a space of functions $F \in \mathcal{F}$, and some additional (calibration) data unseen by the model $Z = (Z_1, \ldots, Z_n)$ where $Z_i = (X_i, F_i)$, construct a prediction set $\mathbb{C}^\alpha \subset \mathcal{F}$ (a set of functions) guaranteed to enclose a next unseen observation F_{n+1} with a user prescribed confidence level: $\mathbb{P}(F_{n+1} \in \mathbb{C}^\alpha) \geq 1 - \alpha$. The space $\mathcal{F}$ has been discretised $F_1 = [F_1(y_1), \ldots, F_1(y_l)] \in \mathbb{R}^l$, but rather finely $l \gg 1$.

1.1 SUMMARY OF THE METHOD

We briefly outline our strategy.

1. Compute $\hat{f}$'s error with respect to the calibration data $e_i = F_i - \hat{f}(X_i)$,
2. Perform a dimension reduction (e.g. an SVD) of e, and project it to a lower dimensional space,
3. Find an enclosing set $\mathcal{Z}$ of the dimension reduced error, $U_i \in \mathcal{Z}$ for all i, and a point p_Z which is close to the error's mode. In this work we use zonotopes for $\mathcal{Z}$,
4. Construct nested prediction regions $\mathcal{Z}^\alpha$ using $\mathcal{Z}$ and p_Z, such that $\mathbb{P}(U_{n+1} \in \mathcal{Z}^\alpha) \geq 1 - \alpha$,
5. Bound truncation error by taking the Cartesian product of the prediction regions $\mathcal{Z}^\alpha$ and a bounding box E of the data of the truncated dimension, $\mathcal{R}^\alpha = \mathcal{Z}^\alpha \times E$,
6. Project $\mathcal{R}^\alpha$ back, and add to the result of the model's prediction $\mathbb{C}^\alpha = \hat{f}(X_{n+1}) + R^\alpha$.

1.2 RELATED WORK

While we review several prior methods, our coverage may not be exhaustive due to the rapidly evolving literature, since uncertainty quantification in Sci-ML is quite a timely problem.

Copula-based conformal prediction: Messoudi et al. [2021] suggest a method to combine univariate prediction sets obtained from conformal prediction using *copulas*, powerful aggregation functions used to decompose multivariate distributions into their marginals and dependencies. This work was further extended by Sun and Yu [2023] to time series. We note however that using copulas to model dependence in high dimensions is challenging, and their proposition may require advanced copula methods, such as vine-copulas, to be applicable to functional surrogates.

Supremum-based conformal prediction: a quite straightforward way to combine univariate conformal predictors is by taking the supremum over a collection of non-conformity scores. Diquigiovanni et al. [2022] take this idea forward by proposing that the scores of each dimension can be normalised or *modulated* by some function $\sigma(t)$:

$$s(x, y) = \sup_{i \in 1, \ldots, N} \left(\sup_{t \in \mathcal{T}} \left| \frac{y_i(t) - \mu(t)}{\sigma(t)} \right| \right),$$

in some prediction region $\mathcal{T}$ of interest, with $\mu(t)$ being the estimated data mean. They suggest that

$$\sigma(t) = \sqrt{\frac{\sum_{i=1}^N (y_i(t) - \mu(t))^2}{N - 1}}$$

is taken to be the estimated data standard deviation. Their method is quite easily adapted to surrogate modelling, by replacing the $\mu(t)$ with a trained predictor $\hat{f}(x_i)(t)$ in both of above expressions, with $\sigma(t)$ now giving the standard deviation of $\hat{f}$'s prediction error. Although this method is simple to apply, we find it can at times give quite wide prediction sets.

Quantile-functional conformal prediction: Ma et al. [2024] propose a quantile neural operator, which they calibrate to give a PAC-style bound on the percentage of points in the function domain that falls within a predicted functional uncertainty set. Our methods diverge as we aim to guarantee the entire function $\mathbb{P}(F_{n+1} \in \mathbb{C}^\alpha) \geq 1 - \alpha$ for all values of F_{n+1} simultaneously. While they control the proportion of the function domain covered (with respect to a uniform sampling of F_{n+1}'s domain), we aim for full-function coverage. Our method also differs as it does not require a quantile function (an additional neural operator, with additional training data) to be trained (but we do require SVDs).

Elliptical-set conformal prediction: Messoudi et al. [2022] propose a multi-target (multivariate) non-conformity score

$$s(x, y) = \sqrt{(y - \hat{f}(x))^\top \hat{\Sigma}^{-1} (y - \hat{f}(x))},$$

where $\hat{\Sigma}$ is sample covariance of the surrogate's prediction error. This non-conformity score has a known analytical

sublevel-set

$$\mathbb{C}^\alpha = \{y \in \mathbb{R}^n \mid s(x,y) \leq q(\alpha)\},$$

which is an ellipsoid centred at $\hat{f}(x)$, and eccentricity related to $\hat{\Sigma}$. They further show their method extends to 'normalised' conformal prediction, where the ellipsoid changes depending on input x.

1.3 CONTRIBUTIONS

Our contribution is most similar to the last of the above methods, where we predict a multivariate set equipped with a guaranteed α-level frequentist performance. Our method diverges as we do not rely on a non-conformity score; instead, we directly construct prediction sets directly on a processed calibration data set. We summarise our contributions:

- A conformal prediction method based on zonotopes
- An application of this technique to functional surrogate models, giving multivariate prediction sets for functional data,
- A method to account for the dimension reduction truncation error, ensuring the guarantees.

2 CONFORMAL PREDICTION AND CONSONANT BELIEF FUNCTIONS

In this section we briefly describe inductive conformal prediction, for calibrating guaranteed prediction sets, particularly used for ML models (but not exclusively), and the related idea of belief functions, which we find to be a useful theory for performing computation using the calibrated sets.

Inductive conformal prediction (ICP) is a computationally efficient version of conformal prediction [Papadopoulos et al., 2002] for computing a set of possible predictions $\mathbb{C}^\alpha : X \mapsto \{\text{subsets of } Y\}$ of an underlying machine learning model $\hat{f} : X \mapsto Y$. The prediction set is equipped with the following probabilistic inequality

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

That is, the probability that the next unobserved prediction Y_{n+1} is in computed set $\mathbb{C}^\alpha(X_{n+1})$ is bounded by $1-\alpha$, where $\alpha \in [0,1]$ is a user-defined error rate. Additionally, Equation 1 can be *guaranteed* if the data is *exchangeable*, i.e. the data used to build $\mathbb{C}^\alpha$ can be replaced with the future unobserved samples Y_{n+1} without changing their underlying distributions.

Most relevant for this paper is how $\mathbb{C}^\alpha$ is constructed, and how inequality 1 is guaranteed. In ICP, one compares the predictions of a pre-trained model $\hat{f}$ to a *calibration* dataset $Z = (Z_1, Z_2, \ldots, Z_n)$ where $Z_i = (X_i, Y_i)$, which is yet unseen by $\hat{f}$. A non-conformity score $s : X \times Y \mapsto \mathbb{R}$ is used to compare $\hat{f}$'s predictions to Z_i, with large values of s indicating a large disagreement between the prediction and the ground truth. A common score used in regression is $s(x,y) = |\hat{f}(x) - y|$. When applied to the calibration data, this yields non-conformity scores $\alpha_i = s(X_i, Y_i)$ for $i = 1, \ldots, n$.

The key insight is that these scores form an empirical distribution, and for a new test point, we can compute a p-value based on this empirical distribution. For a candidate prediction y at test input x, the p-value is defined as the proportion of calibration scores that are at least as large as the test score:

$$p(x,y) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[s(x,y) \leq \alpha_i], \tag{2}$$

where $\mathbb{I}$ is the indicator function. Under exchangeability, this p-value has the property that $\mathbb{P}(p(X_{n+1}, Y_{n+1}) \leq \epsilon) \leq \epsilon$ for any $\epsilon \in [0,1]$.

The prediction set is then constructed by collecting all candidate values y whose p-values exceed the significance level α:

$$\mathbb{C}^\alpha(x) = \{y \in Y \mid p(x,y) > \alpha\}.$$

Equivalently, this can be expressed using a quantile of the empirical distribution of non-conformity scores: taking $q = \alpha_{\lceil (1-\alpha)(n+1) \rceil}$ (the $(1-\alpha)$-quantile of the empirical distribution), we have $\mathbb{C}^\alpha(x) = \{y \in Y \mid s(x,y) \leq q\}$. Note that the prediction regions form a nested family of sets w.r.t α: $\mathbb{C}^{\alpha_1} \supseteq \mathbb{C}^{\alpha_2}$ for $0 \leq \alpha_1 \leq \alpha_2 \leq 1$.

Our framework Computing the level-set of a complicated $s(x,y)$ function may be challenging, so often we opt for quite simple non-conformity scores with known level-sets, e.g., the level-set of $|y - \hat{f}(x)|$ is simply $[\hat{f}(x) \pm q]$.. A challenge which is exacerbated for multivariate problems, somewhat limiting application. We therefore take a different approach, and directly construct $\mathbb{C}^\alpha$ sets using a parametric nested family of sets from the calibration data, a method originally suggested by Gupta et al. [2022]. This allows us to design prediction sets tailored to our particular data, and well-suited for multivariate problems. Using an interpretation of $\mathbb{C}^\alpha$ as belief functions, we can perform additional computations (for example linear and non-linear transformations) on $\mathbb{C}^\alpha$, while still maintaining the guarantee 1. This framework however comes with its own challenges. In some sense we are doing the reverse of conformal prediction, where our challenge is not to compute the level-set $\mathbb{C}^\alpha = \{y \in Y \mid s(x,y) \leq q(\alpha)\}$, rather we begin with $\mathbb{C}^\alpha$ and need to find the membership values $\alpha_i = \sup\{\alpha \in [0,1] \mid Z_i \in \mathbb{C}^\alpha\}$ of the calibration data. Depending on what set-representation is used, computing the membership values can be a costly operation.

Belief functions Cella and Martin [2022] make a useful connection between conformal prediction and belief func-

tions [Shafer, 1976], a generalisation of the Bayesian theory of probability where one can make set-valued probabilistic statements, such as inequality 1 given by conformal prediction.

Belief functions, also called random-sets or Dempster-Shafer structures, are set-valued random variables whose statistical properties (such their cdf, moments, sample realisations, and probability measure) are set-valued. Belief functions form a bound on a collection of partially unknown random variables, often used in robust risk analysis. In particular, the conformal nested prediction sets $\mathbb{C}^\alpha$ can be related a consonant belief function [Dubois and Prade, 1990], and under this framework transformations $f : X \mapsto Y$ of the imprecise random variable are quite simply:

$$\mathbb{C}^\alpha_Y = f(\mathbb{C}^\alpha_X),$$

that is, for each α-level, a single set-propagation of $\mathbb{C}^\alpha_X$ through f is required to determine $\mathbb{C}^\alpha_Y$, maintaining the same α-confidence level. This is comparatively simpler than other representations, which we use to transform fitted $\mathbb{C}^\alpha_X$ through computations (SVDs in particular).

3 ZONOTOPE PREDICTION SETS

Before proceeding to functional data, we must describe how we will construct our multivariate prediction sets $\mathbb{C}^\alpha$. We will use *zonotopes*—a class of convex sets offering advantageous properties for high-dimensional settings, including closure under linear transformations and Minkowski sums. Zonotopes generalise intervals, boxes, hyper-rectangles, and all their rotations, and may be represented on the computer in a compact manner. An n-dimensional zonotope is completely characterised by a vector in $\mathbb{R}^n$ (centre), and a collection of $p \in \mathbb{N}$ vectors in $\mathbb{R}^n$ (generators).

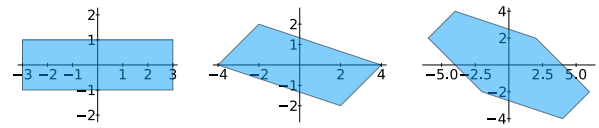
Definition 1 (Zonotope) *Given a centre $c_\mathcal{Z} \in \mathbb{R}^n$ and $p \in \mathbb{N}$ generator vectors in a generator matrix $G_\mathcal{Z} = [g_1, \dots, g_p] \in \mathbb{R}^{n\times p}$, a zonotope is defined as*

$$\mathcal{Z} = \left\{ x \in \mathbb{R}^n \,\middle|\, x = c_\mathcal{Z} + \sum_{i=1}^{p} \xi_i g_i, \;\; \xi_i \in [-1, 1] \right\}$$

We will use the shorthand notation $\mathcal{Z} = \langle c_\mathcal{Z}, G_\mathcal{Z} \rangle$.

Zonotopes can equally be characterised as the image of the box $[-1, 1]^p$ by an affine transformation $T(X) = c_\mathcal{Z} + G_\mathcal{Z} X$. Some example of 2D zonotopes and their generators are shown below (centre is omitted as it corresponds to a translation).

We will use zonotopes as a basis to construct nested prediction regions, giving the probabilistic guarantee 1, calibrated with respect to a particular dataset. For this, we specify a parametric family of nested zonotopes, parameterised by a



$$\begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 3 & 1 \\ -1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 3 & 1 & -2 \\ -1 & 1 & 2 \end{pmatrix}$$

Figure 1: Example zonotopes in blue, and their associated generator matrices immediately below them.

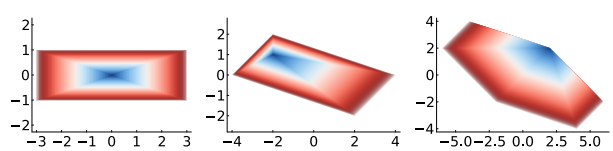
value $\alpha \in [0, 1]$. We find that the following parameterisation is simple, and yields nested sets.

Proposition 1 (nested zonotopic sets) *Given a zonotope $\mathcal{Z} = \langle c_\mathcal{Z}, G_\mathcal{Z} \rangle$, a point $p_z \in \mathcal{Z}$, the following collection of zonotopes are nested*

$$\mathcal{Z}^\alpha_{p_\mathcal{Z}} = \langle \, c_\mathcal{Z}(1-\alpha) + p_\mathcal{Z}\alpha, \; G_\mathcal{Z}(1-\alpha) \, \rangle,$$

with $\alpha \in [0, 1]$. $\mathcal{Z}^\alpha_{p_\mathcal{Z}}$ are nested in the sense that $\mathcal{Z}^{\alpha_1}_{p_\mathcal{Z}} \supseteq \mathcal{Z}^{\alpha_2}_{p_\mathcal{Z}}$ for any $\alpha_1 \leq \alpha_2$.

A detailed proof that $\mathcal{Z}^\alpha_{p_\mathcal{Z}}$ forms a nested collection of sets in found in appendix A.2. **A quick sketch of the proof** is that we show that the half-spaces $H = \{x \in \mathbb{R}^p | a \top x \leq b\}$ composing the box $[-1, 1]^p$ are nested $H^{\alpha_1} \supseteq H^{\alpha_2}$ for $\alpha_1 \leq \alpha_2$ under the transformation $c_\mathcal{Z}(1-\alpha) + p_\mathcal{Z}\alpha + G_\mathcal{Z}(1-\alpha)X$. We also note that $\mathcal{Z}^{\alpha=0}_{p_\mathcal{Z}} = \mathcal{Z}$, (the largest set in the family), and $\mathcal{Z}^{\alpha=1}_{p_\mathcal{Z}} = \{p_\mathcal{Z}\}$ Several examples of parametric nested zonotopes are shown in Figure 2.



$$p_\mathcal{Z} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad p_\mathcal{Z} = \begin{pmatrix} -2 \\ 1 \end{pmatrix} \qquad p_\mathcal{Z} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

Figure 2: Examples of nested zonotopes families, with $\mathcal{Z}$ from Figure 1 and indicated cores $p_\mathcal{Z}$.

Why zonotopes Although Proposition 1 is simple (and its proof not technically deep), the choice of zonotopes is motivated by two key properties:

- **Closure under Cartesian products.** The Cartesian product of zonotopes (or with boxes) yields another zonotope. We use this property in step 5 above (section 1.1) when bounding the truncation error of an SVD.
- **Efficiency under linear transformations.** Zonotopes can be projected exactly through linear maps via a

matrix multiplication. This makes step 6 above very efficient, where we map uncertainty sets back to the prediction space of the surrogate.

Together, these properties make zonotopes a scalable and tractable choice for constructing prediction sets in high-dimensional and function-valued cases.

Note that the zonotopes $\mathcal{Z}_{p_Z}^{\alpha}$ are not yet calibrated to anything, so the desired property $\mathbb{P}(X \in \mathcal{Z}_{p_Z}^{\alpha}) \geq 1 - \alpha$ does not yet hold. The α is so far just a parameter, and needs to be related to a random variable X of interest. In Section 3.2 we explore a method to probabilistically calibrate $\mathcal{Z}_{p_Z}^{\alpha}$ using conformal prediction. This essentially boils down to finding a monotonic function $s : [0,1] \mapsto [0,1]$ for α such that the inequality 1 is guaranteed. Indeed, the $\mathcal{Z}_{p_Z}^{\alpha}$ shown in Figure 2 show a uniformly evaluated α. If one replaces this with a monotonic function $s(\alpha)$, quite dramatically different nested structures are obtained. Examples are shown in Figure 3. One may interpret $s(\alpha)$ as the cumulative probability distribution (cdf) of α, which when sampled yields a random zonotope.

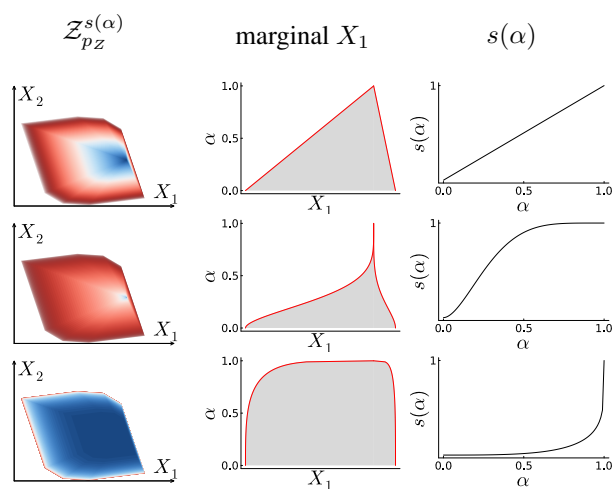


Figure 3: Three examples of the same nested zonotope family $\mathcal{Z}_{p_Z}^{\alpha}$, but with different $s(\alpha)$ functions. The central column shows the projection of the sets onto the X_1 dimension.

We note that inequality 1 can be guaranteed irrespective of the chosen $\mathcal{Z}$ and p_Z, however the 'quality' of the prediction sets (their size relative to their α confidence-level) depends on how well $\mathcal{Z}_{p_Z}^{\alpha}$ captures X's distribution shape. We therefore require the *shape* of the $\mathcal{Z}_{p_Z}^{\alpha}$ sets be *fitted* with respect to the dataset. That is given, some data $\{X_1, \ldots, X_N\}$ sampled from an unknown distribution $X_i \sim F_X$, we would like to find an enclosing zonotope such that all $X_i \in \mathcal{Z}$. We also require the core p_Z to be fitted, which plays the role of defining the region of the highest confidence, the 'central' point of the dataset in some sense. That is, all the prediction sets will contract towards this point, and so it is desirable for the region around this point to occupy a high density of samples. This point can be determined in terms of *data depth*, how deep a point is in a dataset with respect to some metric. This point is somewhat analogous to the Bayesian posterior mode. Methods to fit $\mathcal{Z}_{p_Z}^{\alpha}$ are described in Section 3.1.

A potentially interesting corollary of proposition 1 are a simpler family of nested hyperrectangles $\mathcal{B}_{p_B}^{\alpha}$, in terms of a centre vector $c_{\mathcal{B}} \in \mathbb{R}^n$, radius vector $r_{\mathcal{B}} \in \mathbb{R}^n$, core $p_B \in \mathcal{B}$, and $\alpha \in [0,1]$.

Corollary 1 *The following family of hyperrectangles are nested*

$$\mathcal{B}_p^{\alpha} = \left\{ x \in \mathbb{R}^n \,\middle|\, |x_i - (1-\alpha)c_i - \alpha p_i| \leq (1-\alpha) r_i \right\},$$

for all $i = 1, \ldots, n$, and with $p \in \mathcal{B}$ and $\alpha \in [0,1]$.

Some analysis is simpler in terms of hyperrectangles, and so one could opt to use this family rather than $\mathcal{Z}_{p_Z}^{\alpha}$.

3.1 FITTING ZONOTOPE PREDICTION SETS

In this section we discuss various methods to fit $\mathcal{Z}_{p_Z}^{\alpha}$ to a dataset $X_i \sim F_X$, that is to find a data-enclosing zonotope $X_i \in \mathcal{Z}$ for all samples, and an estimated point p_Z with a high data-depth.

Fitting a rotated hyperrectangle for $\mathcal{Z}$ A simple but effective method is to enclose X_i is using rotated hyperrectangle, with generators $G_{\mathcal{Z}}$ along the principal components of the data.

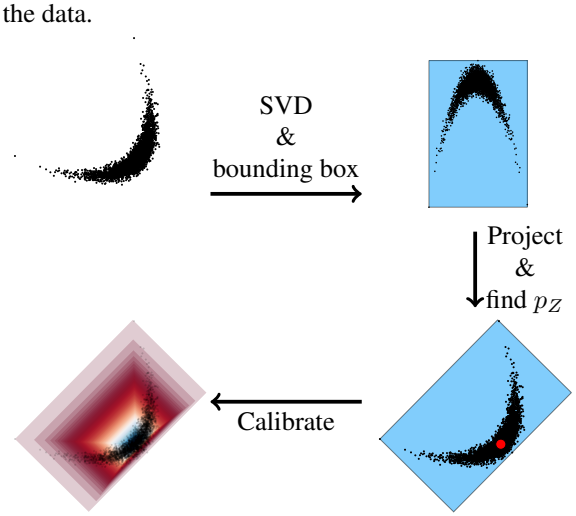


Figure 4: Illustration of fitting a zonotope using the principal component analysis. The data is rotated into the PCA basis, a bounding hyperrectangle is fit in this space, then mapped back to the original coordinates. The resulting zonotope has generators aligned with the principal components, capturing the dominant directions of variation.

A singular value decomposition (SVD) of the data $X = U\Sigma V^{\top}$ yields a diagonal matrix Σ of (decreasing) singular

values and $V^\top$ matrix of singular (or eigen-) vectors. The singular values provide a natural ranking of each eigenvector's contribution to the variance of X. An enclosing hyperrectangle can be easily found by computing the data $\min$ and $\max$ over each dimension of U. Upon converting this hyperrectangle (more details appendix B.1) to a zonotope representation, the zonotope $\mathcal{Z}_U = \langle c_U, G_U \rangle$ can be projected back to the original space by $\mathcal{Z}_X = \langle V\Sigma c_U, V\Sigma G_U \rangle$. This yields a zonotope whose generators are aligned with the eigenvectors of the dataset. Figure 4 illustrates this on a half-moon dataset. Although simple, quick to compute, and scalable, an obvious downside to this method is that it only yields zonotopes with generators as the same number of dimensions of X.

Overapproximating a convex hull for $\mathcal{Z}$ A second method to fit $\mathcal{Z}$ is by first computing the convex hull C_H of X_i, giving a bounding polytope of the data. An enclosing zonotope $\mathcal{Z} \supseteq C_H$ can then be computed (detailed in appendix B.2). This yields a zonotope with half the number of generators as there are bounding half-spaces of C_H. We note that the overapproximation of a convex hull with a zonotope requires a conversion between the vertex representation (v-rep) of polytope to the half-space representation (h-rep), where the v-rep corresponds to a vector of extreme points, and the h-rep is a vector of bounding half-spaces. The conversion from v-rep to h-rep (and vice-versa) can be an expensive operation. Figure 5 illustrates this method on a correlated Gaussian distribution.

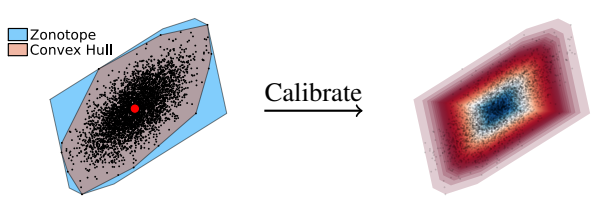


Figure 5: Illustration of fitting using the convex hull.

Additionally, computing C_H can be expensive for higher dimensions or for large quantities of data. For these situations, we recommend the first method, which scales very favorably.

Euclidean data-depth for p_Z A fast, but potentially inaccurate, method is to take p_Z to be the data point X_i with the greatest Euclidean depth w.r.t the sample mean μ

$$p_Z = \arg\max_{X_i} (1 + d_{\mathrm{E}}(X_i))^{-1}$$

where d_{E} is the Euclidean distance of x to μ

$$d_{\mathrm{E}}(x) = \sqrt{(x-\mu)^\top (x-\mu)}.$$

Mahalanobis data-depth for p_Z A straightforward improvement is to include the data covariance Σ in the depth estimation, using the Mahalanobis distance

$$d_{\mathrm{M}}(x) = \sqrt{(x-\mu)^\top \Sigma^{-1} (x-\mu)},$$

with again p_Z being taken to be the sample X_i with the greatest depth

$$p_Z = \arg\max_{X_i} (1 + d_{\mathrm{M}}(X_i))^{-1}.$$

We find this method also fast, scalable, and more accurate than the Euclidean depth when the data covariance can be inverted. Messoudi et al. [2022] use a metric similar to d_{M} as a non-conformity score, the level-sets of which are elliptical.

Approximate Tukey's depth for p_Z A popular measure of depth is Tukey's depth (also known as half-space depth), which for a particular point X_i is defined as the smallest number of points from the dataset that can be contained in any half-spaces passing through X_i. I.e. what is the smallest data partition that can be obtained

$$d_T(x) = \inf_{v \in \mathbb{R}^d} \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}\{v^\top (X_j - x) \geq 0\},$$

where $\mathbb{I}$ is the indicator function that the sample X_i is in the half-space defined by vector v and point x. We then pick p_Z to be the point with the greatest depth

$$p_Z = \arg\max_{X_i} d_T(X_i).$$

Although Tukey's depth is robust, it is highly expensive to compute (requiring 2 loops over the data), we therefore find this approximate Tukey's depth performs quite well up to moderate dimensions

$$\tilde{d}_T(x) = \inf_{v \in V} \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}\{v^\top (X_j - x) \geq 0\},$$

where are V are the normal vectors of the data enclosing set. That is, one only checks the half-spaces composing the enclosing zonotope $\mathcal{Z}$.

3.2 CALIBRATING ZONOTOPE PREDICTION SETS

Calibration with a known distribution For mostly illustrative purposes, we begin by showing how $Z^\alpha_{p_Z}$ can be calibrated from a known multivariate distribution, whose cdf F_X or density f_X are available. This could perhaps be useful for some applications, but the next method (calibrating from sample data) is likely to find wider use. Given a zonotope $\mathcal{Z}$ (ideally containing the range of f_X or otherwise some large probability region ($\mathbb{P}_X(\mathcal{Z}) \approx 1$ if X is unbounded) and a p_Z ideally near the mode, the structure

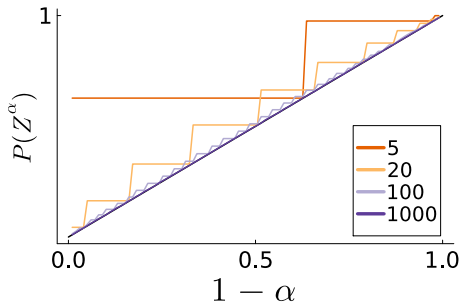


Figure 6: Effect of grid size in membership computation (equation 4). The example computes the prediction sets of a 2-dimensional standard normal gaussian. A large 2M calibration points and 200M points for empirical coverage $\mathbb{P}(\mathcal{Z}^\alpha)$ are used to isolate the effect of the discretisation.

can be calibrated by integrating the density f_X in the sets $\mathcal{Z}^\alpha_{p_Z}$:

$$s(\alpha) = 1 - \int_{\mathcal{Z}^\alpha_{p_Z}} f_X(x)dx.$$

The condition

$$\mathbb{P}(X \in \mathcal{Z}^{s(\alpha)}_{p_Z}) \geq 1 - \alpha$$

holds straightaway.

Calibration from sample data Our method is similar to conformal prediction, as described in section 2, however without a trained regressor $\hat{f}$ and our data $\{X_1, X_2, \ldots, X_n\}$ is multidimensional. We also have a parametric set family $\mathcal{Z}^\alpha_{p_Z}$ in contrast to a non-conformity scoring function. However, the set-membership of the data $\alpha_i = \sup\{\alpha \in [0,1] \mid X_i \in \mathcal{Z}^\alpha_{p_Z}\}$ can be used to rank or score the data. Under the assumption of exchangeability, the probability of another X_{n+1} having a membership score as extreme as the α_i's is equation 2. This directly leads to the (conservative) quantile $s(\epsilon) = \alpha_{\lceil \epsilon n \rceil}$ with sorted α producing a set with the property

$$\mathbb{P}(X_{n+1} \in \mathcal{Z}^{s(\epsilon)}_{p_Z}) \geq 1 - \epsilon, \tag{3}$$

where $\epsilon \in [0,1]$ is a user defined confidence value.

We note that computing the exact membership score $\alpha_i = \sup\{\alpha \in [0,1] \mid X_i \in \mathcal{Z}^\alpha_{p_Z}\}$ may be challenging, as the set must be varied with α until $X_i \notin \mathcal{Z}^\alpha_{p_Z}$, which could be relatively well performed using mathematical optimisation for certain set representations. We however suggest a simple method, where α is uniformly discretised in a grid in $A = \{0, 0.111, \ldots, 0.999, 1\}$, and computing

$$\alpha_i = \max\{\alpha \in A \mid X_i \in \mathcal{Z}^\alpha_{p_Z}\}. \tag{4}$$

Although the exact supremum isn't found, this still gives conservative results, as two collection of scores $\alpha_i \leq \alpha_j$ will yield a lower bound on probabilistic bound on 3. I.e. larger prediction set $\mathcal{Z}^{\alpha_i}_{p_Z} \supseteq \mathcal{Z}^{\alpha_j}_{p_Z}$. This claim is evidenced with a numerical experiment in Figure 6, where one can observe that irrespective of the discretisation a guarantee can be obtained, however the "resolution" of the sets is effected: the more grid points used, the tighter the bounds are. We suggested that the discretisation of the α values is as least a large as the data set, to avoid multiple repetitions of the same α_i values as much as possible. Also note that like in conformal prediction, the maximum number of unique prediction sets that can be obtained is the number of calibration data points provided. Figure 7 gives three examples of calibrating $\mathcal{Z}^\alpha_{p_Z}$ using this method with different data lengths.

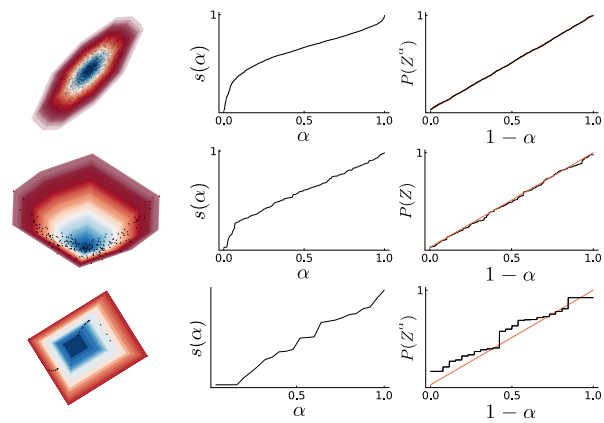


Figure 7: Shows three examples of calibration using conformal prediction. The top row shows a correlated Gaussian of 2000 points, and the middle a 200 samples of a skewed half-moon, and bottom 25 samples of a sin function. The final column shows the empirically tested coverage.

Some potential alternative frameworks exist for reliably calibrating $\mathcal{Z}^\alpha_{p_Z}$, including inferential models [Martin and Liu, 2015], confidence-structures [Balch, 2012], frequency-calibrated belief functions [Denœux and Li, 2018], and scenario theory [De Angelis et al., 2021]. There are likely strong links between these methods and what is proposed here.

4 FUNCTIONAL PREDICTION SETS

Given a pre-trained model $\hat{f} : X \mapsto F$, which maps from Euclidean space $X \in \mathbb{R}^m$ (space of PDE inputs) to a space of functions $F \in \mathcal{F}$ (PDE solutions), discretised on a fine grid $F_1 = [F_1(y_1), \ldots, F_1(y_l)] \in \mathbb{R}^l$, we can compute the error of $\hat{f}$'s error on some unseen data $\{(X_1, F_1), \ldots, (X_n, F_n)\}$ with $e_i = F_i - \hat{f}(X_i)$. We may then reduce the dimension of e_i using an SVD: $e = U\Sigma V^\top$, where the singular values indicate the variance contribution of each eigenvector to the variance of e_i. We may truncate the dimensions of the data, capturing some high (often 99%) of the overall variance, and projecting the data e_i onto these

remaining dimensions, a low dimension representation of $\hat{f}$'s error: U_i. Depending on the size of these dimensions, one of the above fitting methods may be applied to find $\mathcal{Z}_{p_Z}^{\alpha}$. We find the 'rotated hyperrectangle' method and the Mahalanobis depth scale well to high dimensions, with the convex hull method being tighter for smaller dimensions. $\mathcal{Z}_{p_Z}^{\alpha}$ may then be calibrated, giving $\mathbb{P}(U_{n+1} \in \mathcal{Z}^{\alpha}) \geq 1-\alpha$. Mapping $\mathcal{Z}_{p_Z}^{\alpha}$ through the linear transformation back to $\mathbb{R}^l$ is straightforward for the underlying zonotope. The probabilistic component α is also straightforward: each zonotope level-set may be propagated independently $g(\mathcal{Z}_{p_Z}^{\alpha})$ for any function g, i.e. perform one transformation for each prediction set, with the α-guaranteed being retained by the set. Note we may perform this since we compute a multivariate prediction set, this would not be the case had each dimension been fit independently.

These nested sets can be added to the prediction of $\hat{f}$, yielding a functional prediction set

$$\mathbb{C}^{\alpha} = \hat{f}(X_{n+1}) + U\Sigma\mathcal{Z}_{p_Z}^{\alpha}.$$

However, the resulting confidence regions are not strictly a guaranteed bound on (X_{n+1}, F_{n+1}), since we have truncated some data variance during the fitting. Although these dimensions weakly effect the variance of F_i, we cannot strictly claim a guaranteed bound. We therefore propose a quite simple method to account for the uncertainty in these dimensions, without including them in the expensive set calibration.

4.1 BOUNDING TRUNCATION ERROR

To maintain rigorous coverage guarantees, we bound the uncertainty in the truncated dimensions by constructing a hyperrectangle E that encloses the projection of all calibration errors onto the discarded SVD modes. Specifically, we compute the element-wise minimum and maximum of the projected errors in the truncated space, forming a bounding box. Upon taking the Cartesian product $\mathcal{R}^{\alpha} = \mathcal{Z}_{p_Z}^{\alpha} \times E$, one obtains a zonotope in high dimensions which contracts in the important directions as α varies, but remains constant in the truncated dimensions. The α-guarantee remains unchanged due to this operation because: 1) the number of data points remains unchanged (only their dimension), and 2) the data is totally bounded (and remains so as α changes) in the extra dimensions. Slightly more formally, the indicator function for a Cartesian product is

$$\mathbb{I}_{\mathcal{Z}\times E}(x_1, x_2) = \mathbb{I}_{\mathcal{Z}}(x_1)\mathbb{I}_E(x_2),$$

and since E is a bounding box for the calibration data, $\mathbb{I}_E$ always returns 'true' during calibration, and thus would not change the scoring in equation 4 had it been included. The truncated dimensions may thus be ignored during calibration. Of course, when empirically testing the coverage of a prediction set constructed this way, both the zonotope membership $\mathbb{I}_{\mathcal{Z}}$ and the bounding box membership $\mathbb{I}_E$ must be considered.

Indeed, there is a slight performance-loss due to this (our probabilistic bound becomes looser), however this can be expected to be quite minor, as these truncated modes contribute minimally to the overall variance of F_i. Our final prediction set becomes

$$\mathbb{C}^{\alpha} = \hat{f}(X_{n+1}) + \mathcal{R}^{\alpha},$$

where $\mathcal{R}^{\alpha} = U\Sigma(\mathcal{Z}_{p_Z}^{\alpha} \times E)$ has been projected back.

Cartesian product of zonotopes The Cartesian product between two zonotopes $\mathcal{Z}_X \subset \mathbb{R}^n$ and $\mathcal{Z}_Y \subset \mathbb{R}^m$ is a zonotope $\mathcal{Z}_{X\times Y} \subset \mathbb{R}^{n\times m}$, whose centre and generator are a simple concatenation of the centres and generators of $\mathcal{Z}_X$ and $\mathcal{Z}_Y$: $c_{X\times Y} = [c_X \; c_Y]$ and $G_{X\times Y} = [G_X \; G_Y]$.

4.2 PDE SURROGATE EXAMPLES

We use `LazySets.jl` [Forets and Schilling, 2021] for the set construction, and `NeuralOperators.jl` [Pal, 2023] and some prior models from literature Gopakumar et al. [2024a] were used for the base Sci-ML models.

To demonstrate the methodology, we construct functional prediction sets on a Fourier neural operator (FNO) for the Burger's equation, trained on data supplementary data provided from Li et al. [2020], which was built by sampling a numerical PDE solver. After taking an FNO model from literature 1025 points we used to train the SVD, leaving 500 for calibrating the prediction sets, and 503 for validation. Additional information about the training setup and the underlying PDEs are provided in appendix D. The predictions from the FNO are discretise onto a grid of 1024 points. A 32-dimensional rotated hyperrectangle was fitted on the important directions the error's SVD, with p_Z found using the Mahalanobis depth. The entire calibration process took 75s on a modern laptop, with most of the computation spent on computing the membership 4 of the calibration data. Once calibrated, computing a prediction is timed at 0.02s. Four realisations of the training dataset are shown in Figure 8, with the ground truth shown in yellow.

We also note that a 1024 dimensional nested zonotopic set is predicted for the output of the FNO, what's shown in Figure 8 is the axis projection of this set. If we inspect the individual axis, the dependence of the field can also be seen captured.

Table 1 gives a more in depth comparison of our method to the supremum-based method (labelled *modulation*) from Diquigiovanni et al. [2022], for various PDEs and models. We note that our method performs favorably in terms of the tightness of the predicted multivariate set (labelled *efficiency*). Standard metrics for efficiency for multivariate

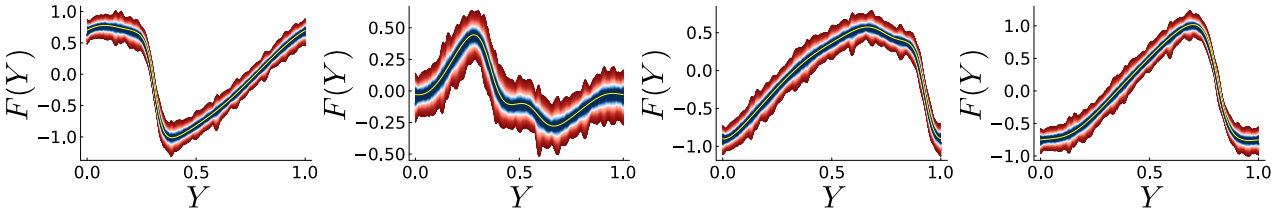


Figure 8: Examples of constructed prediction sets of the FNO for Burger's equation, predicted on test data set. The method predicts nested zonotopes in the high-dimensional space of the surrogate model's prediction. This plot shows the axis-projection of these zonotopes. The ground truth is shown in yellow.

Table 1: Comparison of empirical coverage and efficiency for various PDEs. Coverage is the empirical frequency with which the true function lies within the predicted set (empirical validation of Eq. 1). Efficiency is a measured as the volume of the prediction set — smaller values indicate tighter predictions for the same coverage level.

$\alpha = 0.1$ $\alpha = 0.2$	Modulation	Rotated Box (ours)	Zonotope (ours)
PDE (model)	Coverages		
Burgers (FNO)	89.36	91.41	88.57
	78.32	83.98	79
Burgers (DeepO)	87.79	86.8	90.22
	76.46	79.15	82.24
Wave (FNO)	86	89.2	90
	78	78.4	81.6
Navier Stokes	86.83	89.17	87.33
(FNO)	75.83	78.67	79.67
	Efficiency		
Burgers (FNO)	3.956	2.927^{-1}	$\mathbf{1.910^{-1}}$
	1.634	2.562^{-1}	$\mathbf{1.867^{-1}}$
Burgers (DeepO)	2.742^{2}	**3.79**	5.787
	1.049^{2}	**2.735**	5.386
Wave (FNO)	3.588^{4}	5.656^{-3}	$\mathbf{5.258^{-3}}$
	3.559^{4}	$\mathbf{4.569^{-3}}$	4.744^{-3}
Navier Stokes	9.251^{4}	$\mathbf{3.423^{1}}$	3.529^{1}
(FNO)	7.040^{4}	$\mathbf{2.441^{1}}$	3.282^{1}

X^Y denotes $X \times 10^Y$

conformal prediction are to use the set's volume [Messoudi et al., 2022]. However, since computing the volume of a high dimensional zonotope is challenging, we elect to use the average 2-dimensional projections. Further details of these benchmarks can be found in appendix C and D, including a comparison with lower dimensional data sets from the Mulan and the UCI repository, using more standard MLPs. We note that several other multivariate conformal prediction method are unable to produce results for these models, as they do not scale to the required dimensions. A comparison with these methods is found the appendix in Table 3 for MLPs. We note that for lower dimensional examples, elliptical set conformal prediction performs best, however our method is still competitive.

5 DISCUSSION

The proposed method is similar to the usual score-based conformal prediction, but where we directly solve for a valid prediction set, instead of taking a level-set of a non-conformity scoring function. Indeed, the membership function of $\mathcal{Z}^{\alpha}_{p_Z}$ is the non-conformity score in conformal prediction Gupta et al. [2022]. If one could find an efficient (perhaps analytical) expression for this membership function, in terms of α and the underlying set-representation, then the calibration procedure would be made much more efficient (corresponding to a simple function evaluation), without the need to check the set-membership of $\mathcal{Z}^{\alpha}_{p_Z}$ in 4, which is a costly part of the proposition. This would also additionally greatly simplify the implementation.

We therefore share many of the (dis-)advantages of conformal prediction, including finite-sample guarantees and being model agnostic. Although we show example of regression problems, we believe our method (perhaps with a different set-representation) could apply to multivariate classification problems.

Limitations Since method requires the same assumption as conformal prediction, namely exchangeability, we therefore suffer similar limitations. The guarantee is lost if the dataset is not exchangeable (for example if it changes over time). We also give provide marginal coverage, rather than the stronger conditional coverage, which is more desirable for surrogate modelling. However, methods for improving conditional coverage estimates [Plassier et al., 2025] could be applicable. Other than the usual limitations from conformal prediction, we additionally require an SVD to be trained, which can require a substantial amount of extra training data, in addition to the extra calibration data required for split conformal prediction.

Finally, we mention that our method would greatly benefit for refined methods for fitting zonotopes to data, to improve the tightness of the fitted set.

Acknowledgements

This project was provided with HPC and AI computing resources and storage by GENCI at IDRIS thanks to the grant 2024-AD010615449 on the *Jean Zay* supercomputer's CSL, V100 and A100 partitions.

References

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, pages 1–9, 2024.

Michael Scott Balch. Mathematical foundations for a theory of confidence structures. *International Journal of Approximate Reasoning*, 53(7):1003–1019, 2012.

Christian Jimenez Beltran, Antonio Vergari, Aretha L Teckentrup, and Konstantinos C Zygalakis. Galerkin meets laplace: Fast uncertainty estimation in neural pdes. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024.

Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian Hadfield, Heidy Khlaaf, Jingying Yang, Helen Toner, Ruth Fong, et al. Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims, 2020. URL `https://arxiv.org/abs/2004.07213`.

C Canuto. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Springer-Verlag, 2007.

Leonardo Cella and Ryan Martin. Validity, consonant plausibility measures, and conformal prediction. *International Journal of Approximate Reasoning*, 141:110–130, 2022.

Dongjin Cho, Cheolhee Yoo, Jungho Im, and Dong-Hyun Cha. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas. *Earth and Space Science*, 7(4): e2019EA000740, 2020.

David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, et al. Towards Guaranteed Safe AI: A Framework for Ensuring Robust and Reliable AI Systems. *CoRR*, 2024.

Marco De Angelis, Roberto Rocchetta, Ander Gray, and Scott Ferson. Constructing consonant predictive beliefs from data with scenario theory. In *International Symposium on Imprecise Probability: Theories and Applications*, pages 357–360. PMLR, 2021.

Thierry Denœux and Shoumei Li. Frequency-calibrated belief functions: review and new insights. *International Journal of Approximate Reasoning*, 92:232–254, 2018.

Jacopo Diquigiovanni, Matteo Fontana, and Simone Vantini. Conformal prediction bands for multivariate functional data. *Journal of Multivariate Analysis*, 189:104879, 2022.

Didier Dubois and Henri Prade. Consonant approximations of belief functions. *International Journal of Approximate Reasoning*, 4(5):419–449, 1990. ISSN 0888-613X. doi: https://doi.org/10.1016/0888-613X(90)90015-T. URL `https://www.sciencedirect.com/science/article/pii/0888613X9090015T`.

Marcelo Forets and Christian Schilling. LazySets.jl: Scalable Symbolic-Numeric Set Computations. *Proceedings of the JuliaCon Conferences*, 1(1):11, 2021. doi: 10.21105/jcon.00097.

Vignesh Gopakumar, Stanislas Pamela, and Debasmita Samaddar. Loss landscape engineering via data regulation on PINNs. *Machine Learning with Applications*, 12:100464, 2023. ISSN 2666-8270. doi: https://doi.org/10.1016/j.mlwa.2023.100464. URL `https://www.sciencedirect.com/science/article/pii/S2666827023000178`.

Vignesh Gopakumar, Ander Gray, Joel Oskarsson, Lorenzo Zanisi, Stanislas Pamela, Daniel Giles, Matt Kusner, and Marc Peter Deisenroth. Uncertainty quantification of surrogate models using conformal prediction, 2024a. URL `https://arxiv.org/abs/2408.09881`.

Vignesh Gopakumar, Stanislas Pamela, Lorenzo Zanisi, Zongyi Li, Ander Gray, Daniel Brennand, Nitesh Bhatia, Gregory Stathopoulos, Matt Kusner, Marc Peter Deisenroth, et al. Plasma surrogate modelling using Fourier neural operators. *Nuclear Fusion*, 64(5):056025, 2024b.

Leonidas J Guibas, An Thanh Nguyen, and Li Zhang. Zonotopes as bounding volumes. In *SODA*, volume 3, pages 803–812. Citeseer, 2003.

Chirag Gupta, Arun K. Kuchibhotla, and Aaditya Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, 127:108496, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108496. URL `https://www.sciencedirect.com/science/article/pii/S0031320321006725`.

Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1994.

Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea

Miele, Karthik Kashinath, and Anima Anandkumar. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive Fourier neural operators. In *Proceedings of the platform for advanced scientific computing conference*, pages 1–11, 2023.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Ziqi Ma, David Pitt, Kamyar Azizzadenesheli, and Anima Anandkumar. Calibrated uncertainty quantification for operator learning via conformal prediction. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=cGpegxy12T.

Ryan Martin and Chuanhai Liu. *Inferential models: reasoning with uncertainty*. CRC Press, 2015.

Soundouss Messoudi, Sébastien Destercke, and Sylvain Rousseau. Copula-based conformal prediction for multi-target regression. *Pattern Recognition*, 120:108101, 2021.

Soundouss Messoudi, Sébastien Destercke, and Sylvain Rousseau. Ellipsoidal conformal inference for multi-target regression. In *Conformal and Probabilistic Prediction with Applications*, pages 294–306. PMLR, 2022.

Cedric Nugteren and Valeriu Codreanu. Cltune: A generic auto-tuner for opencl kernels. In *2015 IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip*, pages 195–202. IEEE, 2015.

Avik Pal. On Efficient Training & Inference of Neural Differential Equations, 2023.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine learning: ECML 2002: 13th European conference on machine learning Helsinki, Finland, August 19–23, 2002 proceedings 13*, pages 345–356. Springer, 2002.

Vincent Plassier, Alexander Fishkov, Victor Dheur, Mohsen Guizani, Souhaib Ben Taieb, Maxim Panov, and Eric Moulines. Rectifying conformity scores for better conditional coverage. *CoRR*, 2025.

Michael Redmond and A Baveja. Communities and crime data set. *UCI Machine Learning Repository*, 2009.

Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9:371–421, June 2008. ISSN 1532-4435.

Sophia Huiwen Sun and Rose Yu. Copula conformal prediction for multi-step time series prediction. In *The Twelfth International Conference on Learning Representations*, 2023.

Joaquín Torres-Sospedra, Raúl Montoliu, Adolfo Martínez-Usó, Joan P Avariento, Tomás J Arnau, Mauri Benedito-Bordonau, and Joaquín Huerta. Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems. In *2014 international conference on indoor positioning and indoor navigation (IPIN)*, pages 261–270. IEEE, 2014.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Yibo Yang and Paris Perdikaris. Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, 394:136–152, 2019.

Yibo Yang, Georgios Kissas, and Paris Perdikaris. Scalable uncertainty quantification for deep operator networks using randomized priors. *Computer Methods in Applied Mechanics and Engineering*, 399:115399, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115399. URL https://www.sciencedirect.com/science/article/pii/S0045782522004595.

Fang Zhou, Q Claire, and Ross D King. Predicting the geographical origin of music. In *2014 IEEE International Conference on Data Mining*, pages 1115–1120. IEEE, 2014.

Guaranteed prediction sets for functional surrogate models (Supplementary Material)

Ander Gray[1] **Vignesh Gopakumar**[2,3] **Sylvain Rousseau**[1] **Sébastien Destercke**[1]

[1]Université de technologie de Compiègne, CNRS, Heudiasyc, France
[2]UK Atomic Energy Authority
[3]University College London, UK

A FURTHER DETAIL ABOUT PROPOSITION 1.

Here we provide additional detail about the proposed nested zonotope family $\mathcal{Z}_{p_Z}^{\alpha}$, and a proof that the family is a nested.

A.1 INTUITION BEHIND THE NESTED FAMILY

Figure 9 gives a visual description of how the nested sets $\mathcal{Z}_{p_Z}^{\alpha}$ are constructed. Given a zonotope $\mathcal{Z}$ and a point $p_Z \in \mathcal{Z}$, the centres of the parametric zonotopes move along the line defined by $c_{\mathcal{Z}}(1-\alpha) + p_Z\alpha$, and additionally the generators are contracted by $(1-\alpha)$. Thus, the zonotopes are translated toward p_Z as α increases, and their size reduces. Figure 9 shows an example of a parametric zonotope being constructed for $\alpha = 0.5$. Note that all the sets in $\mathcal{Z}_{p_Z}^{\alpha}$ have the same shape, and only differ by a translation and a contraction. This is partially because all generators are scaled by the same magnitude. Note: $c_{\mathcal{Z}}$ does not need to be inside all zonotopes, however the point p_Z is inside all zonotopes. Indeed, it is the *only* point that is in all zonotopes (given $\mathcal{Z}_{p_Z}^{\alpha=1} = \{p_Z\}$).

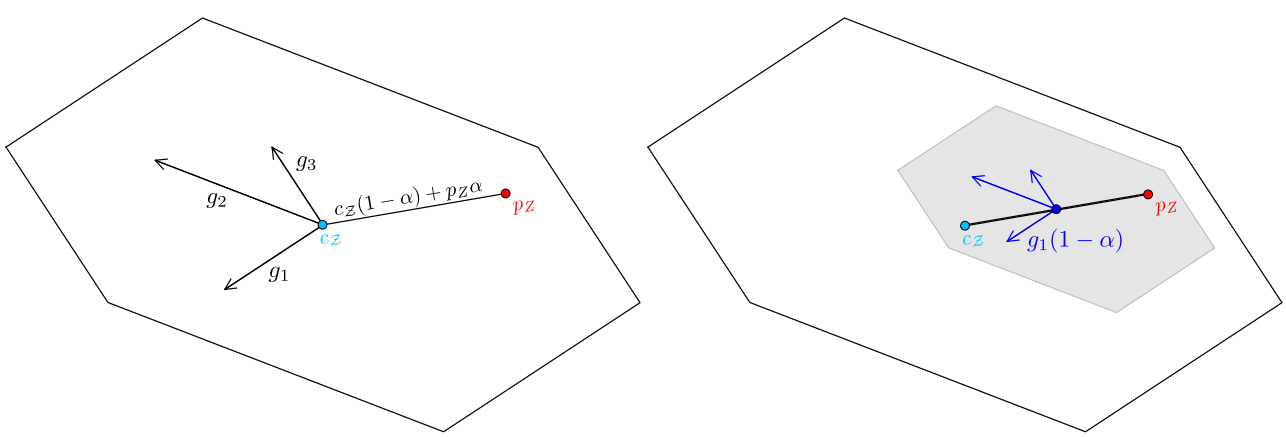


Figure 9: (Left) Shows the outline of an example $\mathcal{Z}$, with generators plotted. Zonotope centre $c_{\mathcal{Z}}$ is in cyan and core p_Z in red, and shows the parametric line defined by $c_{\mathcal{Z}}(1-\alpha) + p_Z\alpha$. (Right) Same outline of $\mathcal{Z}$, additionally with the zonotope defined at $\alpha = 0.5$ in grey, which has its centre halfway in between $c_{\mathcal{Z}}$ and p_Z, and its generators scaled by 0.5.

A.2 PROOF OF PROPOSITION 1.

Here we provide a detailed proof of proposition 1. (nested zonotopic sets), that $\mathcal{Z}_{p_Z}^{\alpha_1} \supseteq \mathcal{Z}_{p_Z}^{\alpha_2}$ for any $0 \leq \alpha_1 \leq \alpha_2 \leq 1$. We remind you that $\mathcal{Z}_{p_Z}^{\alpha}$ is defined as

$$\mathcal{Z}_{p_Z}^{\alpha} = \left\{ x \in \mathbb{R}^n \middle| \; x = c_{\mathcal{Z}}(1-\alpha) + p_Z\alpha + \sum_{i=1}^{p} \xi_i g_i(1-\alpha), \;\; \xi_i \in [-1,1] \right\},$$

for some point $p_Z \in \mathcal{Z}$ and $\alpha \in [0,1]$. The proof is based on the following rational:

1. Take two sets L and K composed as the intersection of n sets $L = \bigcap_{i=1}^{n} A_i$ and $K = \bigcap_{i=1}^{n} B_i$, if each $B_i \subseteq A_i$, then $K \subseteq L$.
2. Since a zonotope can be considered as an intersection of half-spaces (H-representation), it suffices to study the subset-hood of the individual half-spaces composing $\mathcal{Z}_{p_Z}^{\alpha_1}$ and $\mathcal{Z}_{p_Z}^{\alpha_2}$.
3. A zonotope $\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle$ can be characterised as the image of the box $\bigtimes_1^n [-1,1]$ in $\mathbb{R}^n$, where n is the number generators, by an affine transformation defined by the rotation and stretching with matrix $G_{\mathcal{Z}}$ and translation with vector $c_{\mathcal{Z}}$.
4. Equivalently it can be defined as the intersection of the same affine transformation of the half-spaces defining the box $\bigtimes_1^n [-1,1]$.
5. A half-space $H_1 = \{x \in \mathbb{R}^n | a_1^\top x \leq b_1\}$ is a subset of another half-space $H_2 = \{x \in \mathbb{R}^n | a_2^\top x \leq b_2\}$, $H_1 \subseteq H_2$ if:
   (a) a_1 and a_2 are positively aligned, i.e., they are parallel and have the same direction: $a_1 \leq \lambda a_2$ for $\lambda \geq 0$.
   (b) $b_1 \leq \lambda b_2$, where λ the same as (a).

Condition (a) is straightforward to prove: $\mathcal{Z}_{p_Z}^{\alpha_1}$ transforms any half-space normal vectors a as

$$(G_{\mathcal{Z}}(1-\alpha_1))^{-\top} a = (1-\alpha_1)^{-1} G_{\mathcal{Z}}^{-\top} a = (1-\alpha_1)^{-1} a',$$

and $\mathcal{Z}_{p_Z}^{\alpha_2}$ as

$$(G_{\mathcal{Z}}(1-\alpha_2))^{-\top} a = (1-\alpha_2)^{-1} G_{\mathcal{Z}}^{-\top} a = (1-\alpha_2)^{-1} a'.$$

Therefore, since $\alpha_1 \leq \alpha_2$:

$$(1-\alpha_1)^{-1} a' \leq (1-\alpha_2)^{-1} a',$$

holds, and therefore half-spaces transformed by $\mathcal{Z}_{p_Z}^{\alpha_1}$ and $\mathcal{Z}_{p_Z}^{\alpha_2}$ are parallel and have the same direction. We additionally know that the normal vector a' is contracted by $\lambda = (1-\alpha_2)/(1-\alpha_1)$ and that $\lambda \in [0,1]$. For condition (b), $\mathcal{Z}_{p_Z}^{\alpha_1}$ transforms any half-space offset b as

$$\begin{aligned} b_1 &= b + a^\top (G_{\mathcal{Z}}(1-\alpha_1))^{-1} (c_{\mathcal{Z}}(1-\alpha_1) + p_z\alpha_1) \\ &= b + (1-\alpha_1)^{-1} a^\top (G_{\mathcal{Z}}^{-1} c_{\mathcal{Z}}(1-\alpha_1) + G_{\mathcal{Z}}^{-1} p_z \alpha_1). \end{aligned}$$

Setting $J_{\mathcal{Z}} = G_{\mathcal{Z}}^{-1} c_{\mathcal{Z}}$ and $K_{\mathcal{Z}} = G_{\mathcal{Z}}^{-1} p_Z$:

$$\begin{aligned} b_1 &= b + (1-\alpha_1)^{-1} a^\top (J_{\mathcal{Z}}(1-\alpha_1) + K_{\mathcal{Z}}\alpha_1) \\ &= b + a^\top (J_{\mathcal{Z}} + \alpha_1(1-\alpha_1)^{-1} K_{\mathcal{Z}}) \\ &= b + a^\top J_{\mathcal{Z}} + \alpha_1(1-\alpha_1)^{-1} a^\top K_{\mathcal{Z}}. \end{aligned}$$

By symmetry, we also know that for $\mathcal{Z}_{p_Z}^{\alpha_2}$:

$$b_2 = b + a^\top J_{\mathcal{Z}} + \alpha_2(1-\alpha_2)^{-1} a^\top K_{\mathcal{Z}}.$$

Inserting these two expressions into our inequality $b_1 \leq (1-\alpha_2)/(1-\alpha_1) b_2$ and simplifying (noting that $a^\top J_{\mathcal{Z}}$ and $a^\top K_{\mathcal{Z}}$ are scalar):

$$\begin{aligned} \alpha_1(1-\alpha_1)^{-1} &\leq (1-\alpha_2)/(1-\alpha_1)\alpha_2(1-\alpha_2)^{-1} \\ \alpha_1 &\leq \alpha_2. \end{aligned}$$

Thus, any half-space H transformed by $\mathcal{Z}_{p_z}^{\alpha}$ has the property $H^{\alpha_1} \supseteq H^{\alpha_2}$ for any $\alpha_1 \leq \alpha_2$, and $\mathcal{Z}_{p_z}^{\alpha_1} \supseteq \mathcal{Z}_{p_z}^{\alpha_2}$ for any $\alpha_1 \leq \alpha_2$, concluding the proof.

B RELATIONSHIPS BETWEEN SET-REPRESENTATIONS USED IN THE PAPER

We find the `LazySets.jl`[1] [Forets and Schilling, 2021] manual, and software docstrings, quite thorough resources for set-representations and their respective overapproximations. Here we summarise some set isomorphisms and overapproximations used the main paper.

B.1 CONVERTING HYPERRECTANGLES TO ZONOTOPES

Hyperrectangles are exactly representable as zonotopes. A hyperrectangle $\mathcal{B} \subset \mathbb{R}^n$ with centre vector $C_\mathcal{B} \in \mathbb{R}^n$ and radius vector $R_\mathcal{B} \in \mathbb{R}^n$, has the same centre in zonotopic representation $C_\mathcal{Z} = C_\mathcal{B}$, and a diagonal generator matrix $G_\mathcal{Z} \in \mathbb{R}^{n \times n}$ with the radius vector along the diagonals $G_\mathcal{Z} = I_n R_\mathcal{B}$, where I_n is the identity matrix in n dimensions.

B.2 OVERAPPROXIMATING A POLYTOPE WITH A ZONOTOPE

With the algorithm initially proposed by Guibas et al. [2003] (section 4.2), here we summarise the version implemented in `LazySets.jl` [Forets and Schilling, 2021]. Further detail can be found therein.

Given a polytope C_H in vertex representation (for example the result of a convex hull of a dataset) with vertices v_k, and some user-selected directions d_k (to which the constructed zonotope's generators will be parallel to), the overapproximation $\mathcal{Z} \supseteq C_H$ can be performed by solving the following linear program:

$$\begin{aligned} \min \sum_{k=1}^{l} \alpha_k \\ \text{s.t.} \\ v_j = c + \sum_{k=1}^{l} b_{kj} d_k \quad \forall j \\ -\alpha_k \leq b_{kj} \leq \alpha_k \quad \forall k, j \\ \alpha_k \geq 0 \quad \forall k. \end{aligned}$$

The resulting zonotope has center c and generators $\alpha_k d_k$. In this work, we take the directions d_k to be the normal vectors of the enclosing half-spaces the initial polytope.

C EXTENDED EXPERIMENT RESULTS

Table 2 gives further information about the data sets, models, and output dimensions for the experiments in this paper. We note that the MLP benchmarks (Bias to Crime) were originally found in Messoudi et al. [2022] for their comparison of multivariate conformal prediction. Table 3 gives an extended presentation of experimental results, including those for lower dimensional MLPs. Note that for the non-PDE benchmarks, the prediction sets' volume was used as an efficiency metric, while for the high dimensional problem we used average 2D areas, as detailed in the main text.

D PHYSICS, SURROGATE MODELS, AND TRAINING

Here we provide extra details about the PDEs, the functional surrogate models, and their training configurations, used in the main paper. The numerical solvers, data and functional surrogates models used for PDE modelling is borrowed from [Gopakumar et al., 2024a].

[1] https://github.com/JuliaReach/LazySets.jl

Table 2: Information about multivariate and PDE datasets

Name	model	Calibration data	dimensions	Source
Bias	MPL	7750	2	Cho et al. [2020]
Music	MPL	350	2	Zhou et al. [2014]
Indoor	MPL	6946	2	Torres-Sospedra et al. [2014]
SGMM	MPL	79728	4	Nugteren and Codreanu [2015]
Crime	MPL	628	18	Redmond and Baveja [2009]
Burgers	FNO	2048	1024	Gopakumar et al. [2024a]
Burgers	DeepONet	2048	1024	Gopakumar et al. [2024a]
Wave	FNO	7000	4096	Gopakumar et al. [2024a]
Navier Stokes	FNO	7000	4096	Gopakumar et al. [2024a]

Table 3: Results of coverage and efficiency comparison on multivariate and PDE data sets.

$\alpha = 0.1$	Coverages					Efficiency				
$\alpha = 0.2$ dataset / PDE (model)	Modulation	Copula	Ellipse	Rotated Box (ours)	Zonotope (ours)	Modulation	Copula	Ellipse	Rotated Box (ours)	Zonotope (ours)
Bias	89.46	89.06	91.13	88.18	91.29	$\mathbf{5.186^{-2}}$	5.478^{-2}	5.418^{-2}	5.460^{-2}	6.280^{-2}
	78.99	79.95	81.15	79.15	82.59	$\mathbf{3.346^{-2}}$	3.508^{-2}	3.357^{-2}	3.525^{-2}	3.913^{-2}
Music	89.71	94.86	96.57	90.86	90.29	$\mathbf{4.927^{-1}}$	5.504^{-1}	7.467^{-1}	6.905^{-1}	6.125^{-1}
	81.71	86.29	86.29	86.29	82.29	$\mathbf{3.235^{-1}}$	3.460^{-1}	4.104^{-1}	5.512^{-1}	4.425^{-1}
Indoor	90.64	90.96	90.18	88.80	91.22	8.105^{-2}	8.655^{-2}	$\mathbf{7.377^{-2}}$	1.217^{-1}	1.377^{-1}
	80.33	81.11	81.03	78.72	81.20	$\mathbf{4.494^{-2}}$	4.580^{-2}	4.523^{-2}	7.108^{-2}	7.473^{-2}
SGMM	89.98	90.71	89.48	89.88	89.85	2.541^{-5}	2.950^{-5}	$\mathbf{1.176^{-9}}$	2.224^{-6}	2.632^{-6}
	80.01	80.93	79.37	79.74	80.28	4.142^{-6}	4.868^{-6}	$\mathbf{2.331^{-10}}$	1.444^{-6}	1.715^{-6}
Crime	92.04	89.17	90.45	87.58	91.08	4.125^{-10}	8.923^{-7}	$\mathbf{1.583^{-22}}$	6.705^{-18}	7.739^{-15}
	85.35	79.30	80.89	78.66	80.89	1.387^{-12}	1.889^{-9}	$\mathbf{4.171^{-25}}$	8.472^{-19}	2.793^{-15}
Burgers (FNO)	89.36	—	—	91.41	88.57	3.956	—	—	2.927^{-1}	$\mathbf{1.910^{-1}}$
	78.32	—	—	83.98	79	1.634	—	—	2.562^{-1}	$\mathbf{1.867^{-1}}$
Burgers (DeepO)	90.92	—	—	92.77	89.65	2.513^{3}	—	—	**5.868**	8.330
	80.96	—	—	83.01	79.69	9.899^{2}	—	—	**4.052**	7.580
Wave (FNO)	86	—	—	89.2	90	3.588^{4}	—	—	5.656^{-3}	$\mathbf{5.258^{-3}}$
	78	—	—	78.4	81.6	3.559^{4}	—	—	$\mathbf{4.569^{-3}}$	4.744^{-3}
Navier Stokes	86.83	—	—	89.17	87.33	9.251^{4}	—	—	$\mathbf{3.423^{1}}$	3.529^{1}
(FNO)	75.83	—	—	78.67	79.67	7.040^{4}	—	—	$\mathbf{2.441^{1}}$	3.282^{1}

X^Y denotes $X \times 10^Y$ and "—" denotes no result

D.1 BURGERS' EQUATION

The Burgers' equation is a partial differential equation often used to model the convection-diffusion of a fluid, gas, or non-linear acoustics. The one-dimensional equation is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial y} = \nu\frac{\partial^2 u}{\partial^2 y},$$

where u defines the field variables, ν the kinematic viscosity, and with y and t being the spatial and temporal coordinates respectively. We define a family of initial conditions as follows:

$$u(y, t = 0) = \sin(\alpha\pi y) + \cos(-\beta\pi y) + \frac{1}{\cosh(\gamma\pi y)},$$

parameterised by $\alpha \in [-3, 3]$, $\beta \in [-3, 3]$, and $\gamma \in [-3, 3]$.

Data set generation A dataset of 2048 (training) + 1000 (calibration) + 1048 (validation) PDE solutions is generated by Latin Hypercube sampling (uniform) the α, β, and γ parameters (thus generating random initial conditions for the PDE), and then solving Burgers' equation using a spectral solver [Canuto, 2007]. Each simulation is run for 500-time iterations with a $\Delta t = 0.0025$ time step and a spatial domain spanning $[0, 1]$, uniformly discretised into 1024 spatial units. The field at the last time point is then saved as the output, and the surrogate's task is to learn the mapping from the initial condition to the fields final state $u(y, 0) \to u(y, t_{\text{end}})$.

D.2 WAVE EQUATION

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right),$$

where u defines the field variable, c the wave velocity, with x, y and t being the spatial and temporal coordinates respectively. The initial conditions are defined as:

$$u(x, y, t = 0) = \exp\big(-\alpha((x - \beta)^2 + (y - \gamma)^2)\big),$$

parameterised by $\alpha \in [10, 50]$, $\beta \in [0.1, 0.5]$, and $\gamma \in [0.1, 0.5]$, with an additional constraint $\frac{\partial u}{\partial t}(x, y, t = 0) = 0$.

Data set generation A dataset of 500 (training) + 1000 (calibration) + 1000 (validation) PDE solutions is generated by Latin Hypercube sampling the α, β, and γ parameters. The wave equation is solved using a spectral solver with leapfrog time discretization and Chebyshev spectral method on tensor product grid [Gopakumar et al., 2023]. Each simulation runs for 150 time iterations with $\Delta t = 0.00667$ across a spatial domain of $[-1, 1]^2$, discretized into 33 spatial units per dimension. The first 80 time instances of each simulation are used for training.

D.3 NAVIER-STOKES EQUATIONS

The Navier-Stokes scenario that we are interested in modelling is taken from the exact formulation in Li et al. [2020], where the viscosity of the incompressible fluid in 2D is expressed as:

$$\frac{\partial w}{\partial t} + u\nabla w = \nu \nabla^2 w + f, \qquad x \in (0, 1),\ y \in (0, 1),\ t \in (0, T) \tag{5}$$

$$\nabla u = 0, \qquad x \in (0, 1),\ y \in (0, 1),\ t \in (0, T) \tag{6}$$

$$w = w_0, \qquad x \in (0, 1),\ y \in (0, 1),\ t = 0, \tag{7}$$

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. [2020] 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. [2020].

D.4 MULTILAYER PERCEPTRONS

Multilayer Perceptrons (MLPs), a fundamental class of neural networks, are composed of sequential layers of neurons that transform input features through learned weight matrices and non-linear activations [Haykin, 1994]. For a given input $x \in \mathbb{R}^{d_{\text{in}}}$, an MLP with L layers computes:

$$\begin{aligned} h_{i+1} &= \sigma(W_i h_i + b_i), \quad i = 0, \ldots, L - 1 \\ h_L &= W_L h_L + b_L \end{aligned} \tag{8}$$

where $h_0 = x$, $W_i \in \mathbb{R}^{d_{i+1} \times d_i}$ and $b_i \in \mathbb{R}^{d_{i+1}}$ are learnable parameters, and σ is a non-linear activation function (in this case, hyperbolic tangent).

D.5 FOURIER NEURAL OPERATORS

Fourier Neural Operators (FNOs), introduced by Li et al. [2020], are specific instance of the Neural Operator (NO) class of ML models, which have shown efficacy in mapping between function spaces. Following a description by Gopakumar et al. [2024b], a NO can be written as a parameterised mapping between function spaces $G_\theta : A \mapsto U$, where G_θ is a neural network parameterised by θ, with three specific architecture elements, which are sequential:

1. **Lifting**: A fully local, point-wise operation that projects the input domain to a higher dimensional latent representation $a \in \mathbb{R}^{d_a} \to \nu_0 \in \mathbb{R}^{d_{\nu_0}}$,

2. **Iterative Kernel Integration**: Expressed as a sum of a local linear operator, and a non-local integral kernel operator, that iterates $\nu_i \to \nu_{i+1}$ for several layers:

$$\nu_{i+1} = \sigma(W\nu_i(x) + \kappa(a;\phi)\nu_i(x)),$$

where W is the learnable linear components and σ is a non-linear activation (as in traditional networks). The kernel κ (with learnable parameters ϕ) characterises a neural network's layer as a convolution, as the following integral with the prior layer's output ν_i:

$$(\kappa(a;\phi)\nu_i)(x) = \int_D \kappa(x, y, a(x), a(y); \phi)\nu_i(y)dy.$$

3. **Projection** Similar to lifting, but with reversed dimensions $\nu_n \in \mathbb{R}^{d_{\nu_n}} \to u \in \mathbb{R}^{d_u}$, where n is the total number of layers.

A Fourier Neural Operator is a specific class of the above, which defines κ with Fourier convolutions:

$$(\kappa(a;\phi)\nu_i)(x) = \mathcal{F}^{-1}(\mathcal{F}(R_\phi) \cdot \mathcal{F}(\nu_i)),$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier and inverse Fourier transform, and R_ϕ is a learnable complex-valued tensor comprising of truncated Fourier modes. In practical application, the network is discretised to a finite number of modes and the discrete FFT is used for $\mathcal{F}$. FNOs are learnable using gradient-based optimisation using automatic differentiation.

D.6 MLP TRAINING

The trained MLP model consists of:

1. **Input layer**: $\mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{256}$
2. **Hidden layers**: 3 layers of $\mathbb{R}^{256} \to \mathbb{R}^{256}$ transformations with tanh activation
3. **Output layer**: $\mathbb{R}^{256} \to \mathbb{R}^{d_{\text{out}}}$

The network was trained using the Adam optimizer with an initial learning rate of 5×10^{-3}, which was reduced by a factor of 0.5 every 50 epochs. Training proceeded for 500 epochs using mini-batches of size 50 and mean squared error loss. Input features were normalized using a min-max scaling strategy applied per variable. The total parameter count for a single input/output dimension is $256(d_{\text{in}} + 1) + 256^2(L - 1) + 256 + d_{\text{out}}(256 + 1)$.

D.7 FNO TRAINING

The trained FNO model consists of

1. **Lifting layer**: a full connected layer $\mathbb{R}^2 \to \mathbb{R}^{64}$ (192 parameters)
2. **Fourier layers**: 4 Fourier layers (69696 parameters each), each consisting of:
   (a) a full connected later $\mathbb{R}^{64} \to \mathbb{R}^{64}$ (4160 parameters)
   (b) Fourier operator kernel $\mathbb{R}^{64} \to \mathbb{R}^{64}$, truncated to 16 modes (65536 parameters)
3. **Projection layer**: two layer fully connected network $\mathbb{R}^{64} \to \mathbb{R}^{125} \to \mathbb{R}^1$, with Gelu (Gaussian Error Linear Unit) activation (8449 paramaters).

The FNO was trained for 500 epochs using the Adams optimiser with stepsize of 10^{-3} and a weight decay of 10^{-4}, on an L_2 loss, and is timed at around 8 minutes on an NVIDIA A100 GPU.