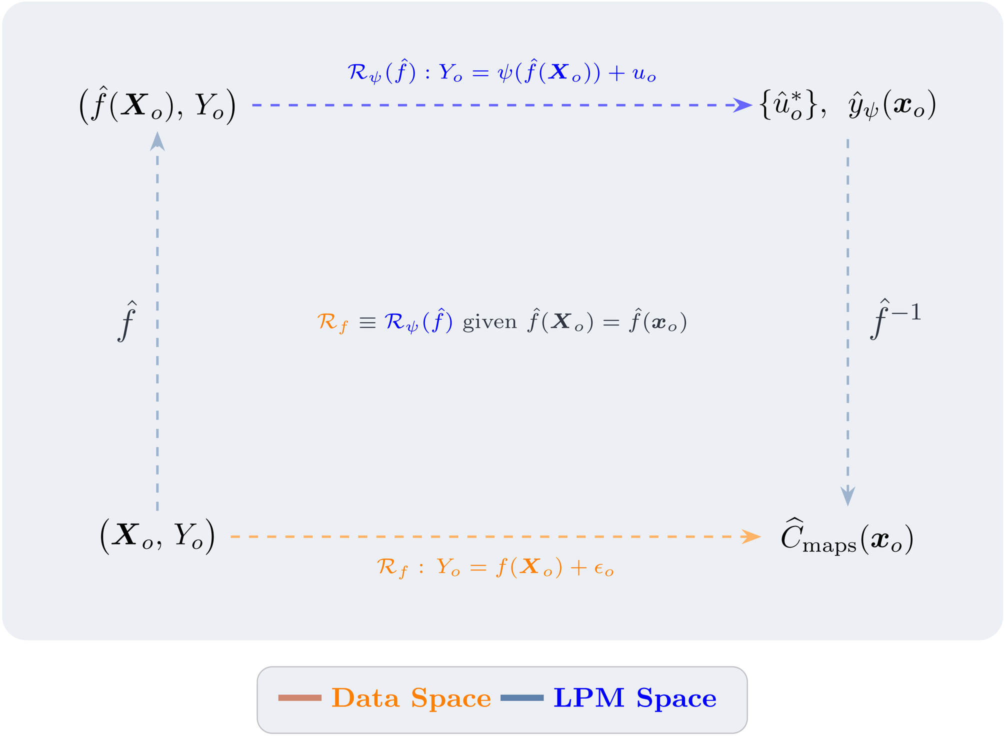
IMPERIAL

The MAPS Algorithm: Fast Model-Agnostic and Distribution-Free Prediction Sets for Supervised Learning

Daniel Salnikov,
Dan Leonte,
Kevin Michalewicz

Motivation

A fundamental problem in modern supervised learning is computing reliable conditional prediction intervals in high-dimensional settings: existing methods often rely on restrictive modelling assumptions, do not scale as predictor dimension increases, or only guarantee marginal (population-level) rather than conditional (individual-level) coverage. We introduce the *lifted predictive model* (LPM), a new conditional representation, and propose the Model-Agnostic Prediction Sets (MAPS) algorithm that produces distribution-free conditional prediction intervals and adapts to any trained predictive model. Our guarantees cover a wide-range of predictive models, including, but not limited to, linear regression, random forests and modern deep neural networks.



Problem Set-Up

Let $(\boldsymbol{X}_i,\, Y_i) \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$, $d \geq 1$, be a a predictor-response pair generated by an unknown joint distribution $P_{\boldsymbol{X},Y}$. In practice, the regression function $f(\boldsymbol{x}) := \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]$ is unknown and is replaced by an estimator $\hat f$ obtained from a learning algorithm applied to the training set. This procedure yields a *pointwise* predictive model:

$$f(\boldsymbol{x}) = \underset{z\in\mathbb{R}}{\operatorname{argmin}}\{\mathbb{E}\left[(Y-z)^2 \mid \boldsymbol{X} = \boldsymbol{x}\right]\}, \quad f : \mathcal{X} \to \mathcal{Y}, \text{ and}$$

$$\hat f \leftarrow \texttt{procedure}\left(\{(\boldsymbol{X}_{i_{\text{train}}},\, Y_{i_{\text{train}}}) : i_{\text{train}} = 1, \ldots, n_{\text{train}}\}\right), \quad \hat f : \mathcal{X} \to \mathcal{Y}.$$

Our objective is computing reliable prediction intervals for an *out-of-sample* Y_o, which are valid for any continuous $\hat f$, that is, for a given confidence level $\alpha \in (0,1)$, we seek a set $\widehat C(\boldsymbol{x}_o) = [\hat c_1(\boldsymbol{x}_o),\, \hat c_2(\boldsymbol{x}_o)] \subset \mathbb{R}$, where $\hat c_1,\, \hat c_2 : \mathcal{X} \to \mathbb{R}$ are "learned" functions with valid conditional coverage:

$$\mathbf{Pr}(Y_o \in \widehat C(\boldsymbol{x}_o) \mid \boldsymbol{X}_o = \boldsymbol{x}_o) \geq 1 - \alpha, \ \ \forall \boldsymbol{x}_o \in \mathcal{X}.$$

Comparison with Alternative Methods

Method	Mod-Ag	Cond	HD Pred	Shape-Free	Class
MAPS (Ours)	✓	✓	✓	✓	✓
MFB (Politis 2015)	✗	✓	✗	✓	✗
SCP (Lei et al. 2018)	✗	✗	✓	✗	✗
DCP (Chernozhukov et al. 2021)	✗	✓	✗	✓	✗

Table 1: Comparison of prediction intervals. **Mod-Ag (Model-Agnostic):** calibrates a trained model to compute adaptive optimal length. **Cond:** valid asymptotic conditional coverage. **HD Pred:** scales with predictor dimension. **Shape-Free:** works for distribution-free heteroscedastic errors. **Class:** provides error bounds for class prediction probabilities.

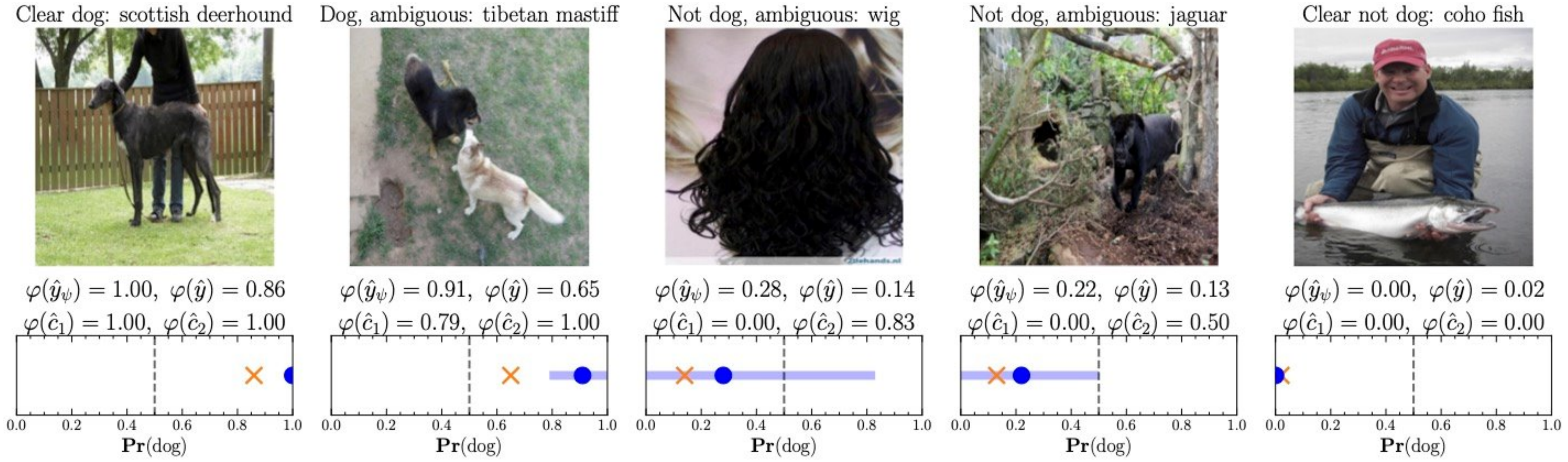


Figure 1: From left to right, prediction intervals for **Pr**(dog) computed with a ConvNeXt (Liu et al. 2022) classifier. Far left, the interval collapses at **Pr**(dog) = 100% for the clear dog image. At the centre, the most ambiguous case and the longest interval. Far right, the interval collapses at **Pr**(dog) = 0% for the clear not dog image. Taken from Salnikov, D., Leonte, D. & Michalewicz, K. (2025), 'The MAPS algorithm: Fast model-agnostic and distribution-free prediction sets for supervised learning', In Submission. pp. 1–28.

A New Algorithm for Computing Prediction Intervals for Arbitrary Predictive Models

The MAPS algorithm generates a sample of lifted residuals and computes the endpoints of the prediction interval at the $(1-\alpha) \times 100\%$ confidence level, that is,

$$\hat c_1,\, \hat c_2 \leftarrow \texttt{maps\_algorithm}(\mathcal{D}_{\text{cal}},\, \hat f,\, \boldsymbol{x}_o,\, \alpha),$$
$$\hat c_1(\boldsymbol{x}_o) = \widehat\psi(\hat f(\boldsymbol{x}_o)) + \widehat q_{\hat u^*}(\alpha/2),$$
$$\hat c_2(\boldsymbol{x}_o) = \widehat\psi(\hat f(\boldsymbol{x}_o)) + \widehat q_{\hat u^*}(1-\alpha/2),$$

where $\widehat\psi$ is a debiasing spline that calibrates pointwise predictions for any $\hat f$, and $\widehat q_{\hat u^*}(\alpha/2)$ and $\widehat q_{\hat u^*}(1-\alpha/2)$ are the $\alpha/2$- and $(1-\alpha/2)$-quantiles of the bootstrap residual sample generated by the MAPS algorithm.

Algorithm 1 MAPS

Require: calibration set $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{X}_{i_{\text{cal}}},\, Y_{i_{\text{cal}}}) : i_{\text{cal}} = 1, \ldots, n_{\text{cal}}\}$, out-of-sample predictor $\boldsymbol{x}_o \in \mathcal{X} \subseteq \mathbb{R}^d$, trained predictive model $\hat f : \mathcal{X} \to \mathbb{R}$, miscoverage level $\alpha \in (0,\, 1)$

1: Apply a nonparametric regression procedure to estimate the lifted regression function ψ by minimising $\mathcal{L}_{\ell pm}$ to get $\widehat\psi$, for example, smoothing splines

$$\widehat\psi = \underset{g\in\mathcal{W}^2}{\operatorname{argmin}}\left\{\frac{1}{n_{\text{cal}}}\sum_{i_{\text{cal}}=1}^{n_{\text{cal}}}\left(Y_{i_{\text{cal}}} - g(\hat f(\boldsymbol{X}_{i_{\text{cal}}}))\right)^2 + \lambda\int (g''(\hat f(\boldsymbol{x}_o)))^2\, dP_{\boldsymbol{X}}\right\}$$

2: Compute $\hat u_{i_{\text{cal}}} = Y_{i_{\text{cal}}} - \hat y_\psi(\boldsymbol{x}_{i_{\text{cal}}})$ and $\hat y_\psi(\boldsymbol{x}_o)$, where $\hat y_\psi(\boldsymbol{x}_o) = \widehat\psi(\hat f(\boldsymbol{x}_o))$

3: Using nonparametric methods, estimate the lifted residual conditional distribution $P_{\hat u|\hat f}$ to get $\widehat P_{\hat u|\hat f}$, for example, kernel density estimators

$$\widehat P_{\hat u|\hat f}(\hat u_o \mid \hat f(\boldsymbol{x}_o)) = \frac{\sum_{i_{\text{cal}}=1}^{n_{\text{cal}}} k_{\hat f}\left([\hat f(\boldsymbol{x}_o) - \hat f(\boldsymbol{x}_{i_{\text{cal}}})]/h_{\hat f}\right) K_u\left([\hat u_o - \hat u_{i_{\text{cal}}}]/h_u\right)}{h_{\hat f}\sum_{i_{\text{cal}}=1}^{n_{\text{cal}}} k_{\hat f}\left([\hat f(\boldsymbol{x}_o) - \hat f(\boldsymbol{x}_{i_{\text{cal}}})]/h_{\hat f}\right)}$$

4: **for** $b = 1$ to $b = B$ **do** ▷ Bootstrap the process to generate lifted residuals

5: Sample $V_o,\, V_i \sim \text{Unif}(0,\, 1)$ for $i = 1, \ldots, n_{\text{cal}}$, and generate pivotal residuals:

$$\hat u_i^{*b} = \widehat P^{-1}_{\hat u_{\text{cal}}|\hat f}(V_i \mid \hat f(\boldsymbol{x}_{i_{\text{cal}}})) \ \text{ and } \ \widehat U_o^{*b} = \widehat P^{-1}_{\hat u_o|\hat f}(V_o \mid \hat f(\boldsymbol{x}_o))$$

6: Bootstrap calibration responses for $i = 1, \ldots, n_{\text{cal}}$ and an out-of-sample one:

$$Y_i^{*b} = \hat y_\psi(\boldsymbol{x}_{i_{\text{cal}}}) + \hat u_i^{*b} \ \text{ and } \ Y_o^{*b} = \hat y_\psi(\boldsymbol{x}_o) + \widehat U_o^{*b}$$

7: Use $\boldsymbol{Y}^{*b} = (Y_1^{*b}, \ldots, Y_{n_{\text{cal}}}^{*b})$ to bootstrap $\widehat\psi$ and the out-of-sample prediction:

$$\widehat\psi^{*b} = \operatorname{argmin}_\psi\{\mathcal{L}_{\ell pm}(\boldsymbol{Y}^{*b},\, \hat f)\} \ \text{ and } \ \hat y_\psi^{*b}(\boldsymbol{x}_o) = \widehat\psi^{*b}(\hat f(\boldsymbol{x}_o)).$$

8: Bootstrap the out-of-sample prediction error:

$$\hat u_o^{*b} = Y_o^{*b} - \hat y_\psi^{*b}(\boldsymbol{x}_o).$$

9: **end for**

Return : $\hat y_\psi(\boldsymbol{x}_o)$, $\{\hat u_o^{*b}\}_{b=1}^B$ ▷ Lifted prediction and residual bootstrap sample

MAPS computes reliable prediction intervals for general neural estimators used in simulation-based inference (SBI)

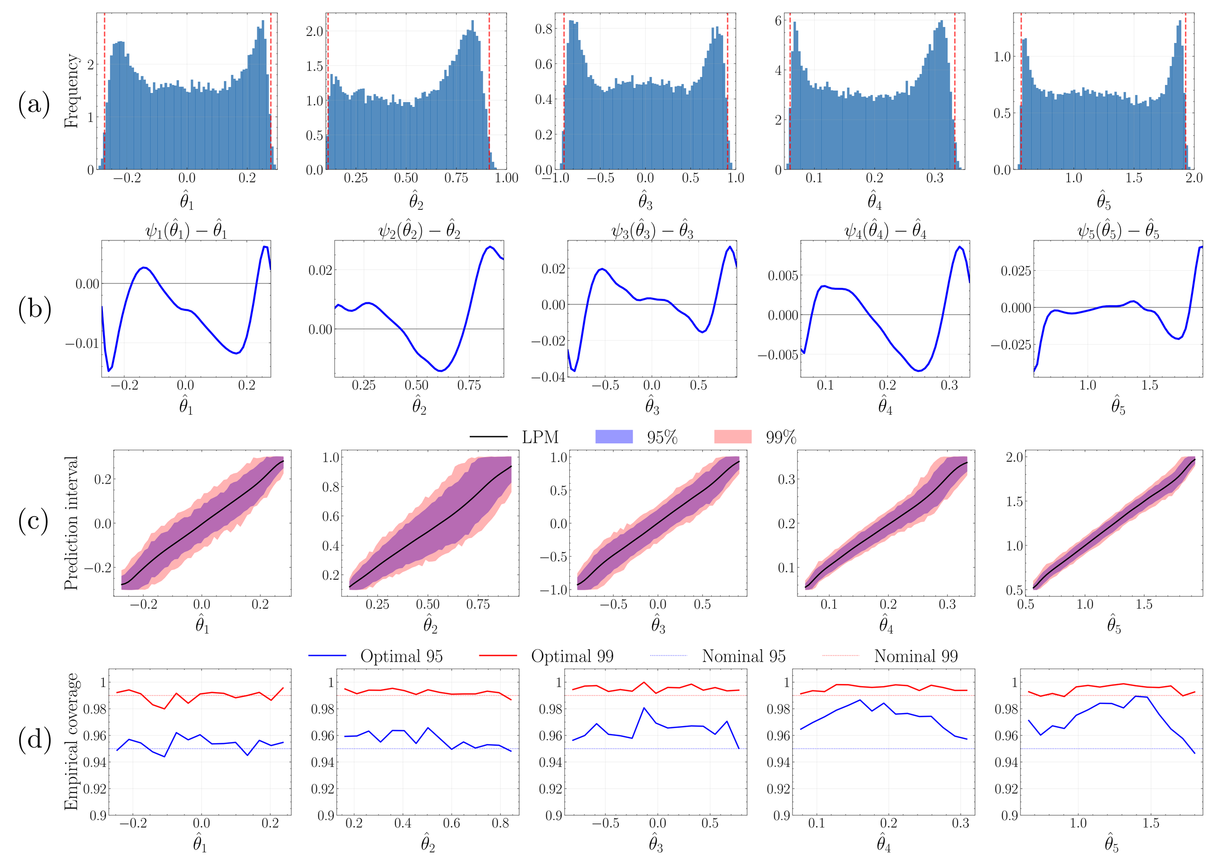


Figure 4: Results for the application in SBI for parameter inference of spatial data under a NMVMN model (Sainsbury-Dale et al., 2025a) using a neural point estimator $\hat{\boldsymbol{\theta}}_j$, where $j = 1, \ldots, 5$. Fig. 4a: Histograms of $\hat{\boldsymbol{\theta}}_j$, where the red lines highlight the 0.5%, 99.5% empirical quantiles. Fig. 4b: LPM bias adjustments $\hat\psi(\hat\theta_j) - \hat\theta_j$. Fig. 4c: 95% and 99% prediction intervals for $\theta_j \mid \hat\theta_j$. Fig. 4d: Empirical coverage of the intervals displayed in Fig. 4c. Fig. 4b–4d are for $\hat{\boldsymbol{\theta}}_j$ in the ranges highlighted by Fig. 4a. See Section 6.1 for details.

Statistical Guarantees

The key insight is that working in the LPM Space is simpler than working in the Data Space. In other words, we exploit the compressions (predictions) to compute lifted prediction intervals, which can be mapped to the data space by conditioning along the contours, say a decompression transform for pointwise predictions in the same contour.

Theorem 1. *Under commonly accepted regularity conditions, $\widehat C_{\text{maps}}(\boldsymbol{x}_o) = [\hat c_1(\boldsymbol{x}_o),\, \hat c_2(\boldsymbol{x}_o)]$ is model-agnostic asymptotically conditionally valid, that is, as $n_{\text{cal}} \to \infty$, we have that*

$$\sup_{\boldsymbol{x}_o}\left\{\mathbf{Pr}\left(Y_o \notin \widehat C_{\text{maps}}(\boldsymbol{x}_o) \mid \hat f(\boldsymbol{X}_o) = \hat f(\boldsymbol{x}_o)\right) - \alpha\right\} = o_{\mathbf{Pr}}(1).$$

Further, if $\hat y_\psi(\boldsymbol{x}_o)$ is asymptotically linear, then for any trained $\hat f$ and $\boldsymbol{x}_o \in \mathcal{X}$,

$$\|P^*_{\hat u_o|\hat f} - P_{u_o|\hat f}\|_\infty = o_{\mathbf{Pr}}(1),$$

where $P^*_{\hat u_o|\hat f}$ is the distribution of $\hat u_o^*$ given the calibration set, $\hat f$ and $\boldsymbol{X}_o = \boldsymbol{x}_o$.

Under two more general conditions and the new notion of contour-homoscedastic errors, our choice of lifted spline estimator is asymptotically linear, hence, we also have that

$$\sup_{\boldsymbol{x}_o\in\hat f^{-1}([y_{\min},\, y_{\max}])}\left\{\mathbf{Pr}\left(Y_o \notin \widehat C_{\text{maps}}(\boldsymbol{x}_o) \mid \boldsymbol{X}_o = \boldsymbol{x}_o\right) - \alpha\right\} = o_{\mathbf{Pr}}(1).$$

If $\|\hat f - f\|_\infty = o_{\mathbf{Pr}}(1)$, $\forall \boldsymbol{x}_o \in \hat f^{-1}([y_{\min},\, y_{\max}])$, then these are also asymptotically optimal,

$$\sup_{\boldsymbol{x}_o\in\hat f^{-1}([y_{\min},\, y_{\max}])}\left\{\lambda_{\text{Leb}}\left(\widehat C_{\text{maps}}(\boldsymbol{x}_o)\Delta C_{\text{ideal}}(\boldsymbol{x}_o)\right)\right\} = o_{\mathbf{Pr}}(1).$$

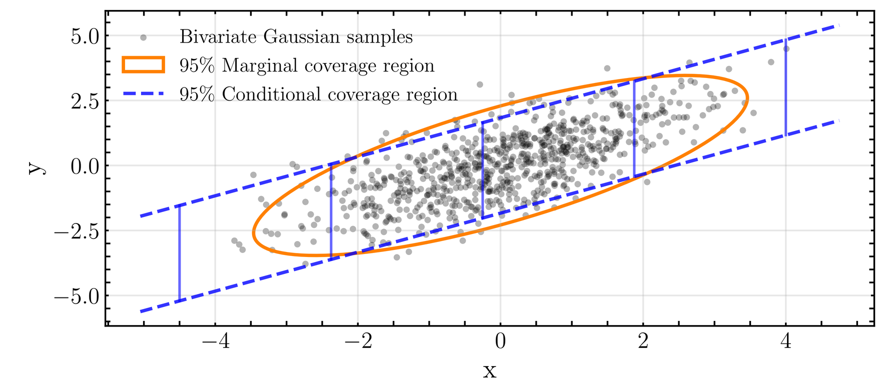


Figure: Comparison of (blue) conditional and marginal (orange) coverage.

Conclusion

- MAPS provides conditional coverage guarantees that do not depend on the predictive model or on proper model specification
- MAPS produces shorter intervals for more accurate models, and converges to the shortest possible ones for consistent (optimal) predictive models
- MAPS is bootstrap consistent for lifted estimators that can be studied as nonparametric regression estimators, for example, our chosen smoothing spline
- MAPS is an easily deployable, computationally efficient method that can be incorporated into machine learning pipelines without altering the underlying methods
- MAPS can be seen as a scanner that wraps around a predictive model to both calibrate it and quantify uncertainty