arXiv:2508.11025v3 [cs.LG] 3 Mar 2026

Zono-Conformal Prediction: Zonotope-Based Uncertainty Quantification for Regression and Classification Tasks

Laura Lützow[1,2] LAURA.LUETZOW@TUM.DE
Michael Eichelbeck[1] MICHAEL.EICHELBECK@TUM.DE
Mykel J. Kochenderfer[2] MYKEL@STANFORD.EDU
Matthias Althoff[1] ALTHOFF@TUM.DE

[1] *School of Computation, Information and Technology, Technical University of Munich, Germany and Munich Center for Machine Learning, Germany*
[2] *Department of Aeronautics and Astronautics, Stanford University, USA*

Abstract

Conformal prediction is a popular uncertainty quantification method that augments a base predictor to return sets of predictions with statistically valid coverage guarantees. However, current methods are often computationally expensive and data-intensive, as they require constructing an uncertainty model before calibration. Moreover, existing approaches typically represent the prediction sets with intervals, which limits their ability to capture dependencies in multi-dimensional outputs. We address these limitations by introducing zono-conformal prediction, a novel approach inspired by interval predictor models and reachset-conformant identification that constructs prediction zonotopes with assured coverage. By placing zonotopic uncertainty sets directly into the model of the base predictor, zono-conformal predictors can be identified via a single, data-efficient linear program. While we can apply zono-conformal prediction to arbitrary nonlinear base predictors, we focus on feed-forward neural networks in this work. Aside from regression tasks, we also construct optimal zono-conformal predictors in classification settings where the output of an uncertain predictor is a set of possible classes. We provide probabilistic coverage guarantees and present methods for detecting outliers in the identification data. In extensive numerical experiments, we show that zono-conformal predictors are less conservative than interval predictor models and standard conformal prediction methods, while achieving a similar coverage over the test data.

Keywords: conformal prediction, interval predictor models, outlier detection, reachset-conformant identification, set-based uncertainty quantification

1 Introduction

Prediction models are increasingly deployed in safety-critical domains, such as autonomous vehicles, healthcare, and robotics. In these domains, models must not only be accurate and robust but also quantify the uncertainty in their predictions to ensure informed decision-making and safety guarantees. Uncertainty quantification methods address this critical need by providing measures of confidence in the outputs of a model, enabling systems to operate more safely under uncertainty.

In this work, we propose a novel set-based method for uncertainty quantification in multi-output prediction models. Our method applies to highly nonlinear models, such as neural networks, and can be used for both regression and classification tasks. Importantly,

©2025 Laura Lützow, Michael Eichelbeck, Mykel J. Kochenderfer, Matthias Althoff.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/.

our approach does not require assumptions about the data distribution while ensuring that the returned prediction sets cover the true output with a user-defined probability.

Traditionally, this problem has been addressed using conformal prediction techniques (Vovk, 2015). However, most conformal prediction approaches require two disjoint data sets: one for identifying an uncertainty model and one for establishing coverage guarantees via calibration. This approach often leads to high computational complexity and requires large amounts of data. In contrast, our method integrates uncertainty modeling and calibration into a single optimization problem using only one data set. To achieve this, we place uncertainties directly into the model of the base predictor and calibrate their sizes using a linear program. This draws inspiration from the interval predictor model framework (Campi et al., 2009), where interval parameters are computed to enclose all data points while minimizing the width of the prediction sets.

Another limitation of many conformal prediction techniques and interval predictor models is their reliance on intervals as prediction sets. While computationally efficient, intervals lack flexibility in representing complex shapes and cannot capture dependencies between multiple outputs. To address this limitation, we employ zonotopes as our set representation. Zonotopes, a class of centrally symmetric polytopes, offer a favorable trade-off between expressiveness and computational efficiency, making them popular in set-based computations for control, verification, and identification tasks. Although we focus on zonotopes, the proposed framework can also accommodate other set representations. Our approach builds on recent developments in reachset-conformant identification (Lützow and Althoff, 2025a), where zonotopic uncertainty sets are identified to ensure that the reachable set of the model contains all system measurements. We extend this framework beyond its traditional application in dynamical systems identification to encompass general prediction models in both regression and classification settings, while providing robust coverage guarantees.

1.1 Related Work

A broad spectrum of research has addressed uncertainty quantification in machine learning (Kabir et al., 2018; Abdar et al., 2021; Cartagena et al., 2021). Probabilistic approaches dominate the field, which typically involve modeling the conditional probability distribution of the outputs using Bayesian methods (Goan and Fookes, 2020) or ensemble-based techniques (Khosravi et al., 2015; Lakshminarayanan et al., 2017; Gal and Ghahramani, 2016). While powerful, these methods often rely on strong assumptions, involve substantial computational costs, and may lack formal coverage guarantees. This work focuses instead on set-based uncertainty quantification methods, which provide prediction sets that contain the true output with a given probability. These approaches typically require fewer assumptions and can offer formal, distribution-free guarantees. One of the most prominent frameworks in this category is conformal prediction, which we review subsequently. We also discuss interval predictor models and reachset-conformant identification, which inspire the proposed zono-conformal prediction approach.

1.1.1 Conformal Predictors (CPs).

Conformal prediction is a distribution-free uncertainty quantification method that transforms point-wise predictions into prediction sets with a user-defined coverage (Vovk et al.,

1999; Lei et al., 2013). Its generality and compatibility with any underlying prediction model have made it popular in modern machine learning. While conformal prediction is well studied for univariate outputs, extending it to multi-output settings remains challenging. The straightforward extension of CPs to multi-dimensional outputs corrects the marginal coverage (e.g., using Bonforonni method) and estimates a prediction interval for each response separately (Stankevičiūtė et al., 2021), resulting in hyper-rectangular and often very conservative prediction sets (Feldman et al., 2023). More advanced methods attempt to reduce this conservatism, but often require large data sets or structural assumptions on the uncertainty.

One line of research explicitly models dependencies between outputs. For example, Messoudi et al. (2021) employ copulas to capture output correlations, and related ideas have been applied to multivariate time-series forecasting (Sun and Yu, 2024). Graph-based deep learning methods have also been explored for quantifying the uncertainty in correlated time-series forecasting (Cini et al., 2025). Another direction focuses on more flexible prediction set shapes. Ellipsoidal prediction sets can be computed by estimating the covariance matrix from the training data (Messoudi et al., 2022; Xu et al., 2024). Gray et al. (2025) uses zonotopes to construct a set of functions that contains the true function with a user-defined probability. Flexible nonconvex shapes can be obtained by training a conditional variational auto-encoder (Feldman et al., 2023) or employing conditional normalizing flows (Fang et al., 2025; Luo and Zhou, 2025). Alternative formulations optimize parameterized shape template functions (Tumu et al., 2024) and prediction regions defined by norm balls (Braun et al., 2025). A comprehensive comparison of multi-output CPs is provided by Dheur et al. (2025).

Most existing approaches decouple the design of prediction set shapes from the calibration step, requiring two separate data sets to guarantee coverage. In this work, we propose a more data-efficient alternative that couples both steps and ensures coverage guarantees using the scenario approach (Calafiore and Campi, 2006). Although conformal prediction and scenario theory have largely developed as separate research directions, recent works (Coppola et al., 2024; Lin and Bansal, 2024; O'Sullivan et al., 2025) highlight their deep connections, showing that conformal prediction guarantees can be recovered within the framework of scenario theory. Relatedly, de Angelis et al. (2021) use scenario theory to construct rectangular predictive sets characterizing the distribution of multivariate data.

1.1.2 Interval Predictor Models (IPMs).

The scenario approach has also been central in establishing probabilistic guarantees for interval predictor models (Campi et al., 2009), which equip simple regression models with prediction intervals. The fundamentals of IPMs lie in interval regression analysis, where interval parameters are optimized through linear programming to ensure that the predicted intervals contain the observed outputs while minimizing interval width (Ishibuchi and Tanaka, 1990). For linear-parameter models, spherical or hyper-rectangular parameter sets are estimated with linear programming, while ellipsoidal parameter sets are identified by solving convex optimization problems (Campi et al., 2009). For nonlinear models, such as single-layer neural networks, parameters can be estimated through incremental nonlinear programming approaches, as each observation introduces non-convex constraints (Campi et al., 2015).

Alternatively, one can first map every input-output pair on a collection of parameter points and then search for a parameter set that tightly encloses parameters corresponding to all scenarios (Crespo et al., 2021). For arbitrary neural networks, an additive uncertainty interval can be directly computed from the maximum prediction error after optimizing the minimax criterion during training (Garatti et al., 2019; Sadeghi et al., 2019). A survey of IPMs is provided by Rocchetta et al. (2021). Furthermore, a comparison with bounded-error approaches, which compute the set of parameters consistent with a predefined error bound and the observed data, is presented in Blesa et al. (2011).

1.1.3 Reachset-conformant Identification.

Uncertainties in dynamical systems can be quantified through reachset-conformant identification. Originally introduced by Althoff and Dolan (2012) and formally defined by Roehm et al. (2016), reachset conformance is a relation between a model and another system, which guarantees the transference of safety properties by ensuring that the model can mimic the behavior of the system (Roehm et al., 2019, Prop. 1). In reachset-conformant identification, reachset conformance for the observed system behavior is established by injecting nondeterminism—here referred to as uncertainties—into the model such that all measurements of the system are reachable. Convex programs can be used to compute zonotopes with minimum size, which enclose the necessary uncertainties for establishing reachset conformance (Gruber and Althoff, 2023), while linear programs can identify enclosing zonotopes, which lead to the smallest reachable sets (Liu et al., 2023; Lützow and Althoff, 2024). The framework has also been extended to nonlinear models using Taylor series expansions (Lützow and Althoff, 2025a) and scaled to larger data sets using recursive algorithms (Lützow and Althoff, 2025b).

1.2 Contributions

In this work, we propose zono-conformal predictors, a novel framework that unifies and generalizes ideas from conformal prediction, IPMs, and reachset-conformant identification. Our contributions are:

- **Generalized framework:** We introduce zono-conformal predictors, which extend IPMs from one-dimensional prediction intervals to multi-dimensional zonotopic prediction sets. In contrast to IPMs, our framework (i) supports multi-output prediction with explicit modeling of dependencies, (ii) efficiently handles nonlinear base predictors via a principled linearization and uncertainty allocation strategy, and (iii) extends to classification tasks.
- **Probabilistic guarantees:** We propose novel methods for detecting outliers in the data and provide probabilistic guarantees on the reliability of the identified predictors.
- **Practical impact:** Zono-conformal predictors can replace multi-output conformal prediction across a wide range of applications. Unlike most existing methods, they require only a small set of calibration data to produce adaptive prediction sets with valid coverage guarantees. Extensive numerical experiments demonstrate that zono-conformal predictors achieve less conservative prediction sets than both classical conformal prediction and IPMs.

Sec. 2 introduces the notation, the necessary background on conformal prediction and IPMs, and zonotopes. Sec. 3 presents the problem statement. Sec. 4 defines zono-conformal predictors and proposes efficient construction techniques for both regression and classification tasks. Sec. 5 outlines three outlier detection strategies and derives probabilistic coverage guarantees for the identified predictors. Sec. 6 compares the conservativeness and coverage of zono-conformal predictors with standard CPs and IPMs in extensive experiments on synthetic and real-world data sets. Additional details on the data sets are provided in Sec. A, and the results of ablation studies are discussed in Sec. B.

2 Preliminaries

This section lays the groundwork for the development of zono-conformal predictors by first introducing the notation used throughout the paper. We then recall two key set-based uncertainty quantification frameworks—conformal prediction and IPMs—which form the conceptual basis for our approach, and explain how coverage guarantees can be derived using scenario theory. Finally, we introduce zonotopes as a flexible and computationally efficient set representation. While reachset-conformant identification is not detailed in this section due to its distinct application domain, its methodological influence on our approach will be highlighted and discussed in subsequent sections.

2.1 Notation

We denote matrices by upper-case letters, vectors and scalars by lower-case letters, and sets by calligraphic letters. Furthermore, we use $\mathbf{1}$, $\mathbf{I}$, and $\mathbf{0}$, respectively, for a vector filled with ones, the identity matrix, and a matrix of zeros with proper dimensions. The j-th element of a vector $b \in \mathbb{R}^o$ is denoted by $b_{(j)}$, while $A_{(\cdot,j)}$ and $A_{(i,j)}$ denote the j-th column and the i-th element of the j-th column of the matrix $A \in \mathbb{R}^{o \times n}$, respectively. Taking the absolute value of each element of the matrix A is denoted by $|A| \in \mathbb{R}^{o \times n}$. The vertical concatenation of the vectors or matrices $A_i \in \mathbb{R}^{o_i \times n}$, $i \in \mathcal{I} \subset \mathcal{N}$, is denoted by $\text{vert}_{i \in \mathcal{I}}(A_i)$. The maximum over all elements of the vector $b \in \mathbb{R}^o$ is denoted by $\max_i(b_{(i)}) \in \mathbb{R}$. Additionally, we use the notation $\text{diag}(b)$ for a diagonal matrix with the elements of the vector b on its main diagonal. The operation $\lfloor x \rceil$ rounds $x \in \mathbb{R}$ to the nearest integer and $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote the floor and the ceil function, respectively. The function $\mathbb{1}\{\text{expression}\}$ returns the value 1 if the boolean expression is true and 0 otherwise. The Minkowski sum of the two sets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathbb{R}^n$ is defined as $\mathcal{X}_1 \oplus \mathcal{X}_2 = \{x_1 + x_2 \mid x_1 \in \mathcal{X}_1,\ x_2 \in \mathcal{X}_2\}$, while the linear transformation of $\mathcal{X}_1$ using the matrix $A \in \mathbb{R}^{o \times n}$ is defined as $A\mathcal{X}_1 = \{Ax \mid x \in \mathcal{X}_1\}$. The set difference of $\mathcal{X}_1$ and $\mathcal{X}_2$ is given by $\mathcal{X}_1 \setminus \mathcal{X}_2 = \{x \mid x \in \mathcal{X}_1 \text{ and } x \notin \mathcal{X}_2\}$.

2.2 Set-Based Uncertainty Quantification

We consider an unknown data-generating process producing samples ($x \in \mathbb{R}^{n_\text{x}}$, $y \in \mathbb{R}^{n_\text{y}}$), without making any assumptions about the underlying probability distribution. Given a calibration data set $\mathcal{M}_\text{cal} = \{(x^{(m)}, y^{(m)})\}_{m=1}^{n_\text{m}}$, both IPMs and CPs generate prediction sets $\mathcal{Y}(\cdot) \subseteq \mathbb{R}^{n_\text{y}}$ such that $\mathcal{Y}(x)$ contains the output y of a new data point (x, y) with a certain probability. We refer to this probability as the coverage $\eta(\mathcal{M}_\text{cal}) = \mathbb{P}\{y \in \mathcal{Y}(x)\}$.

2.2.1 Interval Predictor Models.

IPMs generate prediction intervals for one-dimensional regression tasks (Campi et al., 2009). The predictor is typically constructed as

$$\mathcal{Y}_{\text{IPM}}(x) = \{u^\top \psi(x) \mid u \in \mathcal{U}\} \subset \mathbb{R},$$

where $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$ are uncertain coefficients and $\psi(\cdot) \in \mathbb{R}^{n_u}$ are fixed basis functions. The uncertainty set $\mathcal{U}$ is parameterized by parameters θ, which are identified such that the calibration points $(x, y) \in \mathcal{M}_{\text{cal}} \setminus \mathcal{M}_{\text{out}}$ satisfy $y \in \mathcal{Y}_{\text{IPM}}(x)$, while minimizing the width of the prediction intervals. The set $\mathcal{M}_{\text{out}} \subset \mathcal{M}_{\text{cal}}$ contains potential outliers, which can be identified using a variety of strategies (Huang et al., 1998; Campi et al., 2009).

A common parametrization uses $\psi(x) = [1 \ \tilde{\psi}(x)^\top]^\top$ with $\tilde{\psi}(x) \in \mathbb{R}^{n_\text{u}-1}$, where $u_{(1)}$ lies in the interval $[-\gamma, \gamma]$ and the remaining coefficients $u_{(i)}$, $i = 2, ..., n_\text{u}$ lie in a ball of radius r centered at c:

$$\mathcal{U} = [-\gamma, \gamma] \times \{\tilde{u} \in \mathbb{R}^{n_\text{u}-1} \mid \|\tilde{u} - c\|_2 \leq r\}.$$

The parameter vector $\theta = [\gamma \ r \ c^\top]^\top$ is then optimized with the linear program (Campi et al., 2009, Problem 1)

$$\underset{\theta}{\text{argmin}} \ \mu r + \gamma \tag{1a}$$

$$\text{s.t. } \forall (x, y) \in \mathcal{M}_{\text{cal}} \setminus \mathcal{M}_{\text{out}}\colon \quad y - c^\top \tilde{\psi}(x) - (r\|\tilde{\psi}(x)\|_2 + \gamma) \leq 0, \tag{1b}$$

$$-y + c^\top \tilde{\psi}(x) - (r\|\tilde{\psi}(x)\|_2 + \gamma) \leq 0, \tag{1c}$$

where μ is a positive weight. Coverage guarantees for IPMs can be derived via scenario theory (see Sec. 2.3).

2.2.2 Conformal Predictors.

In this work, we focus on split conformal prediction, the most widely used variant of conformal prediction (Angelopoulos and Bates, 2023). Conformal prediction provides a framework for constructing prediction sets that achieve the marginal coverage

$$\mathbb{E}\{\eta(\mathcal{M}_{\text{cal}})\} \geq 1 - \epsilon. \tag{2}$$

To build such sets, one first defines a score function $s(x, y)\colon \mathbb{R}^{n_\text{x}} \times \mathbb{R}^{n_\text{y}} \to \mathbb{R}$, which quantifies how poorly a data point (x, y) can be explained by a point predictor $f\colon \mathbb{R}^{n_\text{x}} \to \mathbb{R}^{n_\text{y}}$. Given the calibration scores $\{s(x^{(m)}, y^{(m)})\}_{m=1}^{n_\text{m}}$, the prediction set for a new input x is constructed as

$$\mathcal{Y}_{\text{CP}}(x) = \{y \mid s(x, y) \leq q_\epsilon\}, \tag{3}$$

where q_ϵ is the $1 - n_{\text{out}}/n_\text{m}$ empirical quantile of the calibration scores and

$$n_{\text{out}} = n_\text{m} - \lceil (n_\text{m} + 1)(1 - \epsilon) \rceil \tag{4}$$

is the number of data points violating $s(x, y) \leq q_\epsilon$.

The choice of score function depends on the task: For regression tasks, a common score function is the p-norm of the prediction error, i.e., $s(x, y) = \|f(x) - y\|_p$ (Braun et al., 2025). The prediction set for this score function becomes the p-norm ball around $f(x)$ with radius q_ϵ. Alternatively, we can construct and calibrate separate models for each output dimension, e.g., using the score function

$$s_{(j)}(x, y) = |f_{(j)}(x) - y_{(j)}| \tag{5}$$

for each output dimension j, resulting in rectangular prediction sets. However, to obtain a valid joint coverage of $1 - \epsilon$, each model must have a marginal coverage of $1 - \frac{\epsilon}{n_y}$ (Stankevičiūtė et al., 2021). More sophisticated score functions can yield more flexible prediction set shapes, but typically require an additional identification data set.

For classification tasks, a popular score function is

$$s(x, y) = 1 - f_{(y)}(x), \tag{6}$$

which computes 1 minus the softmax output $f_{(y)}(x)$ of the true class y (Angelopoulos and Bates, 2023).

2.3 Scenario Approach

The scenario approach (Calafiore and Campi, 2006) provides a rigorous probabilistic framework for optimization under uncertainty. It replaces uncertain constraints with a finite set of sampled scenarios and gives explicit, non-asymptotic guarantees on constraint satisfaction. These guarantees align conceptually with the Probably Approximately Correct (PAC) learning bounds from statistical learning theory, as both provide finite-sample generalization guarantees that depend on model complexity and the number of available samples (Rocchetta et al., 2024).

Let $\mathcal{M}_{\text{cal}} = \{\delta^{(1)}, \dots, \delta^{(n_{\text{m}})}\} \subset \Delta^{n_{\text{m}}}$ denote n_{m} independent samples from an unknown, time-invariant distribution. We further allow a subset $\mathcal{M}_{\text{out}} \subseteq \mathcal{M}_{\text{cal}}$ of n_{out} samples to be removed, e.g., to account for outliers. A convex scenario program is

$$\underset{\theta}{\operatorname{argmin}}\ J(\theta) \tag{7a}$$

$$\text{s.t. } \forall \delta \in \mathcal{M}_{\text{cal}} \setminus \mathcal{M}_{\text{out}} \colon\ g(\delta, \theta) \leq 0, \tag{7b}$$

where J and g are convex in the parameters $\theta \in \mathbb{R}^{n_\theta}$. We assume that (7) is non-degenerate and admits a unique optimal solution (Rocchetta et al., 2021, Assumptions 1-2). Since the program depends on the sampled data set $\mathcal{M}_{\text{cal}}$, the optimal solution $\theta^*(\mathcal{M}_{\text{cal}})$ is a random variable. We quantify its reliability through the random variable $\eta \colon \Delta^{n_{\text{m}}} \to [0, 1]$ defined by

$$\eta(\mathcal{M}_{\text{cal}}) = \mathbb{P}\{g(\delta, \theta^*(\mathcal{M}_{\text{cal}})) \leq 0\},$$

which represents the probability that a new sample δ satisfies the constraints of (7) for the solution $\theta^*(\mathcal{M}_{\text{cal}})$ and is referred to as coverage probability in the uncertainty quantification

setting. A fundamental result of scenario theory guarantees that the target coverage $1-\epsilon \in (0,1)$ is achieved with a probability of at least (Rocchetta et al., 2021, Thm. 3)

$$\mathbb{P}\{\eta(\mathcal{M}_{\text{cal}}) \geq 1-\epsilon\} \geq 1 - \binom{n_{\text{out}}+n_\theta-1}{n_{\text{out}}} \sum_{i=0}^{n_{\text{out}}+n_\theta-1} \binom{n_{\text{m}}}{i} \epsilon^i (1-\epsilon)^{n_{\text{m}}-i}. \tag{8}$$

The expected value of η can be bounded by (O'Sullivan et al., 2025)

$$\mathbb{E}\{\eta(\mathcal{M}_{\text{cal}})\} \geq 1 - \binom{n_{\text{out}}+n_\theta-1}{n_{\text{out}}} \frac{n_{\text{out}}+n_\theta}{n_{\text{m}}+1}. \tag{9}$$

Scenario theory thus provides a natural framework for analyzing the reliability of prediction models. Specifically, the linear program for constructing IPMs in (1) can be interpreted as a convex scenario program with $\delta = (x, y)$. Similarly, CP construction can also be formulated as a scenario program with optimization variable $\theta = q_\epsilon$, cost function $J(q_\epsilon) = q_\epsilon$, and constraint function $g(x, y, q_\epsilon) = s(x, y) - q_\epsilon$. Consequently, the formal coverage guarantees in (8) and (9) hold for prediction sets generated by both IPMs and CPs, without any assumptions on the underlying data distribution. Moreover, (9) reduces to (2) whenever $\epsilon \geq \binom{n_{\text{out}}+n_\theta-1}{n_{\text{out}}} \frac{n_{\text{out}}+n_\theta}{n_{\text{m}}+1}$, which holds for conformal predictors with $n_\theta = 1$ and n_{out} defined as in (4).

2.4 Zonotopes

In contrast to IPMs and most CPs, we represent the prediction sets using zonotopes, which are centrally symmetric convex polytopes.

Definition 1 (Zonotopes (Kühn, 1998)) *A zonotope $\mathcal{Z} \subset \mathbb{R}^n$ can be described by a center vector $c \in \mathbb{R}^n$ and a generator matrix $G \in \mathbb{R}^{n \times \nu}$:*

$$\mathcal{Z} = \left\{ c + \sum_{i=1}^{\nu} \lambda_i G_{(\cdot,i)} \;\middle|\; \lambda_i \in [-1, 1] \right\} = \langle c, G \rangle.$$

Since zonotopes are closed under the set operations linear transformation and Minkowski sum and omit a compact representation size, they are used for efficient set-based computations in control (Schäfer et al., 2024), reachability analysis (Girard, 2005; Althoff et al., 2007), state estimation (Alamo et al., 2003; Combastel, 2015), or identification (Chabane et al., 2014). The size of a zonotope can be characterized using the interval norm:

Definition 2 (Interval Norm (Althoff, 2023)) *The interval norm for the zonotope $\mathcal{Z} = \langle c, G \rangle$ is defined as the absolute sum over all elements of G:*

$$\|\mathcal{Z}\|_I = \mathbf{1}^\top |G| \mathbf{1}.$$

3 Problem Statement

We introduce zono-conformal predictors (ZCPs), which return multi-dimensional zonotopic prediction sets $\mathcal{Y}_{\text{ZCP}}(x) \subseteq \mathbb{R}^{n_{\text{y}}}$ that contain the true output with a certain probability. The model $\mathcal{Y}_{\text{ZCP}}(\cdot)$ is built from the data $\mathcal{M}_{\text{cal}} = \{(x^{(m)}, y^{(m)})\}_{m=1}^{n_{\text{m}}}$ by ensuring each prediction

set $\mathcal{Y}_{\text{ZCP}}(x^{(m)})$ contains the corresponding output $y^{(m)}$. At the same time, we minimize a proxy for the volume of the prediction sets using the function size$(\cdot)$, evaluated for inputs $\{x_{\text{eval}}^{(m)}\}_{m=1}^{n_{\text{eval}}}$, which are representative samples from the set of admissible inputs. Minimizing a proxy for the volume of the prediction sets ensures that the predictions are not too conservative and, thus, informative. For standard regression tasks, where the observed outputs $y^{(m)}$ correspond to continuous values, the problem can be formalized as follows:

Problem 1 (Uncertainty Quantification for Regression) *Given a regression task with data* $\mathcal{M}_{\text{cal}} = \{(x^{(m)}, y^{(m)})\}_{m=1}^{n_{\text{m}}}$*, we want to find a model* $\mathcal{Y}_{\text{ZCP}}(\cdot)$ *that solves*

$$\min_{\mathcal{Y}_{\text{ZCP}}(\cdot)} \sum_{m=1}^{n_{\text{eval}}} \text{size}(\mathcal{Y}_{\text{ZCP}}(x_{\text{eval}}^{(m)})) \tag{10a}$$

$$\text{s.t. } \forall m \in \{1, \dots, n_{\text{m}}\}\colon \; y^{(m)} \in \mathcal{Y}_{\text{ZCP}}(x^{(m)}). \tag{10b}$$

Furthermore, we propose to use ZCPs in classification tasks. Here, the observed output $y^{(m)} \in \{0,1\}^{n_{\text{y}}}$ is a binary vector, whose i-th element has the value 1 if $x^{(m)}$ can be classified to the i-th class and 0 otherwise. We assume that the output of the available point predictor $f(x) \in \mathbb{R}^{n_{\text{y}}}$ is a vector, indicating a classification to the i-th class, if the i-th element of $f(x)$ is bigger than or equal to all other elements. A non-deterministic classifier, on the other hand, returns a prediction set $\mathcal{Y}_{\text{ZCP}} \subset \mathbb{R}^{n_{\text{y}}}$, which can encode multiple classes. By introducing the functions

$$\text{classes}(\mathcal{Y}) = \left\{i \mid \exists y \in \mathcal{Y}\colon y_{(i)} = \max_j y_{(j)}\right\} \text{ and } \text{classes}(y) = \left\{i \mid y_{(i)} = \max_j y_{(j)}\right\}, \tag{11}$$

which map a set $\mathcal{Y}$ or a vector y onto the encoded classes, the uncertainty quantification problem for classification tasks can be stated as follows:

Problem 2 (Uncertainty Quantification for Classification) *Given a classification task with data* $\mathcal{M}_{\text{cal}} = \{(x^{(m)}, y^{(m)})\}_{m=1}^{n_{\text{m}}}$*, we want to find a model* $\mathcal{Y}_{\text{ZCP}}(\cdot)$ *that solves*

$$\min_{\mathcal{Y}_{\text{ZCP}}(\cdot)} \sum_{m=1}^{n_{\text{eval}}} \text{size}(\mathcal{Y}_{\text{ZCP}}(x_{\text{eval}}^{(m)})) \tag{12a}$$

$$\text{s.t. } \forall m \in \{1, \dots, n_{\text{m}}\}\colon \; \text{classes}(y^{(m)}) \subseteq \text{classes}\left(\mathcal{Y}_{\text{ZCP}}(x^{(m)})\right). \tag{12b}$$

Since Problem 1 and Problem 2 are complex nonlinear programming problems, we will introduce some additional assumptions on the construction of $\mathcal{Y}_{\text{ZCP}}(\cdot)$ in the following section so that both problems can be solved with linear programming.

Additionally, we want to evaluate the coverage probability of ZCPs. To cater to cases where less conservative prediction sets are required while lower coverage is sufficient, we also describe outlier detection methods that discard anomalous data points.

4 Zono-Conformal Prediction

In this section, we discuss how we can efficiently construct zono-conformal predictors $\mathcal{Y}_{\text{ZCP}}(\cdot)$ from observed data. We propose the following procedure:

1. **Deterministic Model**: We start with a deterministic prediction model, whose output $f(x) \in \mathbb{R}^{n_\text{y}}$ can be described by a nonlinear function of the input $x \in \mathbb{R}^{n_\text{x}}$.

2. **Uncertainty Placement**: We create an augmented function $\tilde{f}(x, u) \in \mathbb{R}^{n_\text{y}}$, with $\tilde{f}(x, \mathbf{0}) = f(x)$, by inserting variables $u \in \mathbb{R}^{n_\text{u}}$ into the function $f(x)$ to model the uncertainty of the predictor.

3. **Uncertainty Quantification**: We identify a zonotopic uncertainty set $\mathcal{U}$ from observed data such that the nondeterministic model

$$\mathcal{Y}_{\text{ZCP}}(x) = \{f(x) + \bar{D}(x)u \mid u \in \mathcal{U}\} \subset \mathbb{R}^{n_\text{y}}, \tag{13}$$

   with $\bar{D}(x) = \nabla_u \tilde{f}(x, u)|_{u=\mathbf{0}}$, solves Problem 1 in the regression setting or Problem 2 in the classification setting.

These steps are explained in detail in the following subsections.

4.1 Deterministic Model

The deterministic prediction model $f(x)$ can be constructed from prior knowledge about the data generation process or identified from additional data. In this work, we assume $f(x)$ is already given. While the uncertainty placement and quantification steps, which are described in the following subsections, can be applied to arbitrary nonlinear models, we will focus on neural network models in the experiments in Sec. 6.

4.2 Uncertainty Placement

In the zono-conformal prediction framework, we aim to construct prediction sets that adapt to the input rather than remaining constant in size and shape. To achieve this, we insert variables $u \in \mathbb{R}^{n_\text{u}}$ into the deterministic function $f(x)$ to obtain the augmented function $\tilde{f}(x, u)$. This augmented function retains the original behavior of the deterministic predictor when $u = \mathbf{0}$, i.e., $\tilde{f}(x, \mathbf{0}) = f(x)$. The variables u, referred to as uncertainties, represent components of the predictor whose effect on the output is uncertain.

Uncertainties can be incorporated at different locations within the predictor. One straightforward approach is to directly influence the output by adding output uncertainties $u_\text{y} \in \mathbb{R}^{n_\text{y}}$. Furthermore, parametric uncertainties $u_\text{p} \in \mathbb{R}^{n_\text{p}}$ can be added in form of uncertain model parameters.

Example 1 *Consider the deterministic predictor* $f(x)\colon \mathbb{R}^2 \to \mathbb{R}^2$,

$$f(x) = \begin{bmatrix} 3x_1 + x_2 \\ 2x_2^3 \end{bmatrix}.$$

Inserting output uncertainties u_y and parametric uncertainties u_p leads, for example, to the augmented function

$$\tilde{f}(x,u) = \begin{bmatrix} (3+u_{p,1})x_1 + (1+u_{p,3})x_2 \\ (2+u_{p,2})\big((1+u_{p,4})x_2\big)^3 \end{bmatrix} + u_\mathrm{y},$$

with $u = [u_\mathrm{p}^\top \; u_\mathrm{y}^\top]^\top$.

Using only output uncertainties corresponds to constant prediction sets, while introducing parametric uncertainties enables adaptive, input-dependent prediction sets. For physics-based or interpretable models, output uncertainties are appropriate when the output is always noisy or uncertain (e.g., due to measurement noise), whereas parametric uncertainties should be added for components subject to unknown disturbances or partially known parameters (e.g., friction coefficients depending on an unobserved environment).

However, for complex models, such as neural networks, the optimal placement of uncertainties is generally unknown. In this case, one must select a subset $u \in \mathbb{R}^{n_u}$ from a large pool of candidate uncertainties $\tilde{u} = [u_\mathrm{p}^\top \; u_\mathrm{y}^\top]^\top \in \mathbb{R}^{n_\mathrm{p}+n_\mathrm{y}}$, with $n_\mathrm{u} < n_\mathrm{p} + n_\mathrm{y}$. Identifying too many uncertainties can lead to increased computational cost and reduced coverage due to overfitting. Conversely, a poorly chosen uncertainty placement strategy can lead to excessive conservatism and, in the worst case, may render the constraints in (10b) or (12b) unsatisfiable. To address this, we propose a simple yet effective strategy:

1. Select all output uncertainties u_y to ensure adequate control over each output dimension.
2. Randomly sample the remaining $n_\mathrm{u} - n_\mathrm{y}$ uncertainties from the pool of parametric uncertainties u_p.

As shown in Sec. B.1, this strategy consistently outperforms purely random selection and a more complex, deterministic approach using QR-factorization.

4.3 Uncertainty Quantification

We construct the zono-conformal predictor as in (13) by evaluating the first-order Taylor-series approximation of $\tilde{f}(x,u)$ at $u = \mathbf{0}$ over the uncertainties $u \in \mathcal{U}$. This linearization is motivated by reachset-conformant identification, where the reachable set of a nonlinear system under bounded uncertainty $u \in \mathcal{U}$ is approximated using a Taylor series (Althoff, 2013). The uncertainty set $\mathcal{U}$ is then identified by solving Problem 1 or Problem 2, both of which reduce to linear programs under the following assumptions:

- To motivate small and informative prediction sets, we minimize the following cost function, which computes the summed interval norms of randomly rotated versions of the prediction set $\mathcal{Y}_\mathrm{ZCP}(x)$:

$$\mathrm{size}(\mathcal{Y}_\mathrm{ZCP}(x)) = \sum_{i=0}^{n_\mathrm{r}} \|R_i \mathcal{Y}_\mathrm{ZCP}(x)\|_I, \tag{14}$$

  where $n_\mathrm{r} \geq 1$ is a user-defined integer, R_i, with $i \geq 1$, are random orthogonal matrices, and $R_0 = \mathbf{I}$. The standard approach in reachset-conformant identification uses only

the unrotated interval norm, i.e., (14) with $n_\mathrm{r} = 0$, but our experiments in Sec. B.2 show that incorporating random rotations significantly reduces the conservativeness of the prediction sets.

- We represent the uncertainty set $\mathcal{U}$ by the zonotope

$$\mathcal{U} = \langle \mathbf{0}, G_\mathrm{u}\mathrm{diag}(\alpha)\rangle, \quad \alpha \in \mathbb{R}^{n_\theta}_{\geq 0}, \tag{15}$$

  where the scaling factors α are the optimization variables, while the unscaled generator matrix $G_\mathrm{u} \in \mathbb{R}^{n_\mathrm{u} \times n_\theta}$ must be specified by the user. As shown by the experiments in Sec. B.1, defining G_u as the identity matrix usually provides good results.

Before deriving the linear program formulations for regression and classification, we present a few results needed for both cases.

Lemma 3 *The prediction set of the zono-conformal predictor in* (13) *with the uncertainty set in* (15) *can be described by the zonotope*

$$\mathcal{Y}_\mathrm{ZCP}(x) = \langle f(x),\ \bar{D}(x)G_\mathrm{u}\mathrm{diag}(\alpha)\rangle. \tag{16}$$

Proof *Inserting* (15) *into* (13)*, results in* (16). ■

Lemma 4 *The interval norm of the rotated prediction set $R_i\mathcal{Y}_\mathrm{ZCP}(x)$ of the zono-conformal predictor in* (13) *with the uncertainty set in* (15) *can be computed as*

$$\|R_i\mathcal{Y}_\mathrm{ZCP}(x)\|_I = \mathbf{1}^\top |R_i\bar{D}(x)G_\mathrm{u}|\alpha, \tag{17}$$

Proof *Using* (16) *and Def. 2, we obtain*

$$\|R_i\mathcal{Y}_\mathrm{ZCP}(x)\|_I = \mathbf{1}^\top |R_i\bar{D}(x)G_\mathrm{u}\mathrm{diag}(\alpha)|\mathbf{1},$$

which is equal to (17) *for $\alpha \geq \mathbf{0}$.* ■

Lemma 5 *A point y is contained in the prediction set $\mathcal{Y}_\mathrm{ZCP}(x)$ of the zono-conformal predictor in* (13) *with the uncertainty set in* (15)*, i.e., $y \in \mathcal{Y}_\mathrm{ZCP}(x)$, iff*

$$\exists \bar{\beta} \colon |\bar{\beta}| \leq \alpha, \quad y = f(x) + \bar{D}(x)G_\mathrm{u}\bar{\beta}. \tag{18}$$

Proof *This result can be derived using Def. 1 and Lem. 3 analogously to Lütz ow and Althoff (2024, Thm. 2).* ■

Thus, we obtain the following formulation for regression problems:

Theorem 6 (Zono-Conformal Regression) *Problem 1 with the cost function in* (14) *can be solved with the zono-conformal predictor given by* (13) *and* (15)*, where the optimal scaling factors* α *are computed with the following linear program:*

$$\underset{\alpha,\beta}{\operatorname{argmin}} \sum_{m=1}^{n_{\text{eval}}} \sum_{i=0}^{n_{\text{r}}} \mathbf{1}^\top |R_i \bar{D}(x_{\text{eval}}^{(m)}) G_{\text{u}}| \alpha \tag{19a}$$

$$\text{s.t. } \forall m \in \{1, \ldots, n_{\text{m}}\}: \quad \mathbf{0} \leq \alpha, \tag{19b}$$

$$\mathbf{0} \leq \alpha + \beta_m, \tag{19c}$$

$$\mathbf{0} \leq \alpha - \beta_m, \tag{19d}$$

$$y_\Delta^{(m)} = \bar{D}(x^{(m)}) G_{\text{u}} \beta_m, \tag{19e}$$

with $y_\Delta^{(m)} = y^{(m)} - f(x^{(m)})$ *and* $\beta = [\beta_0^\top, \ \ldots, \ \beta_{n_{\text{m}}}^\top]^\top$ *is a vector of additional optimization variables to enforce the containment constraints.*

Proof *Evaluating the interval norm of* $\mathcal{Y}_{\text{ZCP}}(x)$ *over the inputs* $x_{\text{eval}}^{(m)}$ *with the constraint* (19b)*, as in* (10a)*, leads to the linear cost in* (19a)*. Constraints* (19c) *to* (19e) *can be obtained by using* (18) *for the containment constraints in* (10b) *and introducing a new optimization variable* β_m *for each observation* $(x^{(m)}, y^{(m)})$*.* ■

For classification problems, we assume that each output y is mapped to a single class using classes(y). To extend zono-conformal predictors to multi-label classification, the calibration data set is constructed by including data points of the form $(x, \mathbf{e}_i)$ for each class i that is a possible label for the input x, where $\mathbf{e}_i$ denotes the i-th unit vector.

Theorem 7 (Zono-Conformal Classification) *Problem 2 with the cost function in* (14) *can be solved with the zono-conformal predictor given by* (13) *and* (15)*, where the optimal scaling factors* α *are computed with the following linear program:*

$$\underset{\alpha,\beta}{\operatorname{argmin}} \sum_{m=1}^{n_{\text{eval}}} \sum_{i=0}^{n_{\text{r}}} \mathbf{1}^\top |R_i \bar{D}(x_{\text{eval}}^{(m)}) G_{\text{u}}| \alpha \tag{20a}$$

$$\text{s.t. } \forall m \in \{1, \ldots, n_{\text{m}}\}: \quad \mathbf{0} \leq \alpha, \tag{20b}$$

$$\mathbf{0} \leq \alpha + \beta_m, \tag{20c}$$

$$\mathbf{0} \leq \alpha - \beta_m, \tag{20d}$$

$$-T_m f(x^{(m)}) \leq T_m \bar{D}(x^{(m)}) G_{\text{u}} \beta_m, \tag{20e}$$

with the new optimization variable $\beta = \text{vert}_{m \in \{1,\ldots,n_{\text{m}}\}}(\beta_m)$ *and*

$$T_m = \text{vert}_{n_{\text{y}}}(\mathbf{e}_{\text{classes}(y^{(m)})}^\top) - \mathbf{I}. \tag{21}$$

Proof *Evaluating the interval norm of* $\mathcal{Y}_{\text{ZCP}}(x)$ *over the inputs* $x_{\text{eval}}^{(m)}$ *with the constraints* (20b)*, as in* (12a)*, leads to the linear cost in* (20a)*. Since we have*

$$y_{(i)} = \begin{cases} 1 & \text{if } i = \text{classes}(y), \\ 0 & \text{otherwise}, \end{cases} \tag{22}$$

the constraint in (12b) *can be written as*

$$\begin{aligned}
\text{classes}(y^{(m)}) \in \text{classes}\big(\mathcal{Y}_{\text{ZCP}}(x^{(m)})\big) &\overset{(11)}{\Leftrightarrow} \exists z \in \mathcal{Y}_{\text{ZCP}}(x^{(m)})\colon \max_j z_{(j)} = z_{(\text{classes}(y^{(m)}))} \\
&\Leftrightarrow \exists z \in \mathcal{Y}_{\text{ZCP}}(x^{(m)})\colon z \leq \text{vert}_{n_y}(\mathbf{e}^\top_{\text{classes}(y^{(m)})})z \\
&\Leftrightarrow \exists z \in \mathcal{Y}_{\text{ZCP}}(x)\colon \mathbf{0} \leq T_m z \\
&\overset{(18)}{\Leftrightarrow} \exists \beta_m\colon |\beta_m| \leq \alpha,\ \mathbf{0} \leq T_m(f(x^{(m)}) + \bar{D}(x^{(m)})G_{\text{u}}\beta_m),
\end{aligned}$$

resulting in (20c) *to* (20e). ■

5 Coverage

In conformal prediction, the trade-off between high coverage and smaller prediction sets can be controlled by selecting how many calibration data points should be contained in the corresponding prediction sets. This principle also extends to ZCPs. We present three methods for identifying outliers, i.e., data points whose removal most effectively reduces the conservatism of ZCPs. Furthermore, we describe the resulting coverage guarantees of ZCPs and compare them with those of CPs.

5.1 Outlier Detection

Finding the subset of data points whose removal yields the greatest reduction in optimization cost is a combinatorial problem, and solving it through exhaustive search is computationally infeasible. In the following sections, we present three efficient methods for outlier detection, which are evaluated and compared in Sec. B.4. All three methods lead to significantly smaller optimization costs compared to a naive outlier detection approach, which is not tailored to the zono-conformal prediction framework.

5.1.1 Search over Boundary Points

In this work, we define a data point (x, y) as a boundary point if there exists at least one $\alpha_{(j)}$, $j = 1, \ldots, n_\theta$, such that an infinitesimal reduction in $\alpha_{(j)}$ causes the output y to fall outside its corresponding prediction set $\mathcal{Y}_{\text{ZCP}}(x)$, as shown in Fig. 1. Boundary points, therefore, induce constraints in the linear programs (19) or (20) that block any further reduction of the optimization cost at the optimum. Leveraging this definition, outlier detection can be framed as a search over boundary points, similarly to (Campi et al., 2009, Alg. 1). Specifically, we construct a search tree rooted at the original problem, where each node represents a modified version in which one data point—a boundary point in the problem of its parent node—has been removed. This process is repeated for n_{out} layers, yielding leaf nodes that each exclude n_{out} data points. The leaf node with the lowest objective value then identifies the optimal set of outliers.

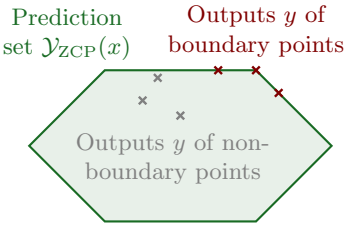


Figure 1: Boundary and non-boundary points.

The runtime of this algorithm critically depends on how quickly one can identify boundary points at each node. By assuming non-degenerate problems (Campi et al., 2009, Def. 4), Campi et al. (2009) identify boundary points as data points whose removal leads to a strict decrease of the optimization cost. However, this approach does not work for degenerate problems and leads to high computational cost for large data sets, as we have to compute the optimization cost for all possible removals. Instead, we propose a more scalable method that identifies the set of boundary points for a given node i and the associated calibration set $\mathcal{M}_{\text{cal},i} = \{(x^{(m)}, y^{(m)})\}_{m=1}^{n_{\text{m},i}}$ directly from the solution of (19) for regression tasks or (20) for classification tasks:

Proposition 8 (Detection of Boundary Points) *Boundary points can be identified by computing the vector δ using the following linear program:*

$$\underset{\beta,\delta}{\operatorname{argmin}} \quad -\mathbf{1}^\top \delta \tag{23a}$$

$$\text{s.t. } \forall m \in \{1, \dots, n_{\text{m},i}\}: \qquad 0 \leq \delta_{(m)} \tag{23b}$$

$$-\alpha^* \leq \beta_m - f\delta_{(m)}, \tag{23c}$$

$$-\alpha^* \leq -\beta_m - f\delta_{(m)}, \tag{23d}$$

$$\text{(19e) } \textit{for regression or } \text{(20e) } \textit{for classification}, \tag{23e}$$

where α^* is the optimal solution of (19) *or* (20)*, and the elements of $f \in \mathbb{R}^{n_\text{u}}$ are given by*

$$f_{(j)} = \begin{cases} 1 & \text{if } \alpha^*_{(j)} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The m-th data point is a boundary point iff $\delta_{(m)} = 0$.

Proof *A data point $(x^{(m)}, y^{(m)})$ is not a boundary point iff this data point satisfies the containment constraint* (10b) *for regression tasks or* (12b) *for classification tasks, even when the non-zero elements of the optimal scaling factors α^* are slightly decreased. This occurs if there exists a vector β_m satisfying the containment constraints* (10b) *via constraints* (19c) *to* (19e) *or* (12b) *via constraints* (20c) *to* (20e) *for $\alpha = \alpha^*$ but whose elements satisfy*

$$\forall j \colon |\beta_{m(j)}| < \alpha^*_{(j)} \quad \textit{or} \quad \alpha^*_{(j)} = 0. \tag{24}$$

The linear program (23) *encodes this condition by maximizing the slack $\delta_{(m)} = \min_{j \in \mathcal{J}}(\alpha^*_{(j)} - |\beta_{m(j)}|)$, which is required to be greater than or equal to zero, with $\mathcal{J} = \{j \mid \alpha^*_{(j)} > 0\}$. The constraints* (23c) *to* (23e) *ensure that each β_m remains feasible for the containment constraints under $\alpha = \alpha^*$. A non-zero $\delta_{(m)}$ implies that there exists a vector β_m satisfying* (24)*, meaning the m-th data point is not a boundary point. Conversely, $\delta_{(m)} = 0$ means no such β_m exists and the data point is a boundary point.* ■

5.1.2 Greedy Search over Boundary Points

To reduce computational costs, a greedy variant of the search strategy described in Sec. 5.1.1 can be employed. Rather than expanding all nodes at each level, the greedy algorithm only retains the child node with the lowest cost, trading optimality guarantees for significantly improved scalability.

5.1.3 Detection via Mixed-Integer Linear Program

Alternatively, we can incorporate the outlier detection into the optimization problems (19) and (20), resulting in the following mixed-integer linear programs (MILPs):

Proposition 9 (Zono-Conformal Regression with Outlier Removal) *If n_{out} data points are permitted to violate the constraints in* (10b)*, Problem 1 can be solved with Thm. 6, where constraint* (19e) *is replaced by the constraints*

$$\mathbf{0} = \bar{D}(x^{(m)})G_{\text{u}}\beta_m - y_{\Delta}^{(m)}\rho_{(m)}, \tag{25a}$$

$$n_{\text{m}} - n_{\text{out}} \leq \mathbf{1}^{\top}\rho, \tag{25b}$$

$$\rho \in \{0,1\}^{n_{\text{m}}}. \tag{25c}$$

The vector $\rho \in \mathbb{B}^{n_{\text{m}}}$ is an additional optimization variable, where $\rho_{(m)}$ is equal to 1 if data point m is required to satisfy the constraint in (10b) *and 0 if the measurement is considered as an outlier and permitted to violate the constraint in* (10b)*.*

Proof *The containment $y^{(m)} \in \mathcal{Y}_{\text{ZCP}}(x^{(m)})$ is enforced via* (19c) *to* (19e)*. By multiplying the left-hand side of* (19e) *with $\rho_{(m)}$, we obtain* (25a)*. If $\rho_{(m)} = 1$, the new constraint* (25a) *is equal to* (19e)*, which means the containment $y^{(m)} \in \mathcal{Y}_{\text{ZCP}}(x^{(m)})$ is enforced. If $\rho_{(m)} = 0$, the constraints* (19c)*,* (19d) *and* (25a) *can be trivially solved with $\beta_m = \mathbf{0}$, which means $y^{(m)}$ is classified as an outlier since the containment $y^{(m)} \in \mathcal{Y}_{\text{ZCP}}(x^{(m)})$ is not enforced.*

The limitation of the values ρ can take is encoded via the integer constraint in (25c)*, while* (25b) *ensures that a maximum of n_{out} measurements are classified as outliers.* ■

Proposition 10 (Zono-Conformal Classification with Outlier Removal) *If n_{out} data points are permitted to violate the constraints in* (12b)*, Problem 2 can be solved with Thm. 7, where constraint* (20e) *is replaced by the constraints*

$$\mathbf{0} \leq T_m\bar{D}(x^{(m)})G_{\text{u}}\beta_m + T_m f(x^{(m)})\rho_{(m)}, \tag{26a}$$

$$n_{\text{m}} - n_{\text{out}} \leq \mathbf{1}^{\top}\rho, \tag{26b}$$

$$\rho \in \{0,1\}^{n_{\text{m}}}. \tag{26c}$$

The vector $\rho \in \mathbb{B}^{n_{\text{m}}}$ is an additional optimization variable, where $\rho_{(m)}$ is equal to 1 if data point m is required to satisfy the constraint in (12b) *and 0 if the measurement is considered as an outlier and permitted to violate the constraint in* (12b)*.*

Proof *Analogously to the proof of Prop. 9.* ■

5.2 Coverage Guarantees

The coverage guarantees of IPMs and CPs can be derived from scenario theory (O'Sullivan et al., 2025), as discussed in Sec. 2.3. Assuming the data points in $\mathcal{M}_{\text{cal}}$ are independently drawn from a time-invariant distribution and independent from the evaluation points $\{x_{\text{eval}}^{(m)}\}_{m=1}^{n_{\text{eval}}}$, we can directly apply these results to ZCPs, since (19) and (20) are convex

scenario programs. Thus, the coverage of ZCPs satisfies (8) and (9). Depending on the application, one may either prescribe a desired coverage level and adjust the number of uncertainty parameters n_θ and outliers n_{out} accordingly, or fix these parameters in advance and derive the resulting coverage.

Although the guarantees in (8) and (9) formally hold regardless of the output dimension, higher-dimensional outputs typically require more complex models with a larger number of parameters n_θ, which leads to smaller theoretical confidence bounds—see right-hand side of (8). Since the number of identified parameters n_θ for most CPs is equal to one, while ZCPs often involve a larger n_θ to capture richer uncertainty structures, we expect higher coverage guarantees for CPs under the same outlier budget n_{out}. Nevertheless, our experiments in Sec. 6 demonstrate that ZCPs consistently yield smaller prediction sets while maintaining comparable empirical coverage, suggesting a favorable trade-off between informativeness and coverage.

6 Experiments

We analyze the performance of ZCPs using different synthetic and real-world data sets. For each task, we train 30 neural networks utilizing tanh activation functions, which will serve as the base predictors $f(x)$. The network architectures as well as the specifications of the data sets are provided in Tab. 1. More details on the synthetic data sets and the *Photovoltaic* data set are provided in Sec. A. Before each training, we randomly select 10.000 data points from each data set with more than 10.000 instances. We randomly split the data into a calibration set, which contains 10% of the instances, a test set, containing 15% of the instances, and a training set for training the base predictor, consisting of the remaining 75% of the instances.

For the construction of ZPMs, we select $n_{\text{u}} = n_{\text{y}} + 0.1n_{\text{p}}$ uncertainties as described in Sec. 4.2, where parametric uncertainties u_{p} are added to all network biases in the hidden layers, we use the cost function from Lem. 4 with $n_{\text{r}} = 10$, and we detect outliers via greedy search over boundary points as described in Sec. 5.1.2. Other uncertainty placement strategies, cost functions, and outlier detection methods are evaluated in Sec. B. Our methods are implemented in the MATLAB toolbox CORA[1].

6.1 Regression Tasks

For regression tasks, we compare ZCPs with the following two baseline methods:

- IPMs that use the zono-conformal prediction framework to handle nonlinear systems with multi-dimensional outputs while representing the uncertainty sets and prediction sets with multi-dimensional intervals. The resulting predictor is identical to a ZCP that uses a diagonal generator matrix for the uncertainty set $\mathcal{U}$ and whose output set is overapproximated by a multi-dimensional interval.

- CPs that use the prediction error as score function—see (5)—to identify a prediction interval for each dimension separately.

1. The CORA toolbox is available at https://tumcps.github.io/CORA/.

Task	n_x	n_y	Network Size	n_p	n_total	Reference
SD-R1	2	2	$[64, 64]$	128	∞	Sec. A
SD-R2	3	2	$[64, 64]$	128	∞	Sec. A
Energy	8	2	$[64, 64]$	128	768	Tsanas and Xifara (2012)
CASP	8	2	$[64, 64]$	128	45730	Feldman et al. (2023)
Housing	17	2	$[64, 64]$	128	21613	Feldman et al. (2023)
BlogPost	279	2	$[64, 64]$	128	52397	Feldman et al. (2023)
SCM20D	61	4	$[64, 64]$	128	8966	Tsoumakas et al. (2011)
SCM1D	280	4	$[64, 64]$	128	9803	Tsoumakas et al. (2011)
RF1	64	8	$[64, 64]$	128	9004	Tsoumakas et al. (2011)
Photovoltaic	49	4	$[64, 256, 64]$	384	8639	Sec. A
SD-C1	2	3	$[64, 64]$	128	∞	Sec. A
SD-C2	3	4	$[64, 64]$	128	∞	Sec. A
Cover type	54	7	$[128, 128]$	256	581012	Blackard and Dean (1999)
MNIST	784	10	$[128, 128]$	256	70000	LeCun et al. (1998)

Table 1: Overview of the different prediction tasks, where the first ten tasks are regression and the last four are classification data sets.

Since the more sophisticated conformal prediction approaches, as reviewed in Sec. 1.1, require additional data or access to the training data to estimate the shape of the uncertainty sets (Cini et al., 2025; Feldman et al., 2023; Messoudi et al., 2022; Xu et al., 2024; Gray et al., 2025; Fang et al., 2025; Luo and Zhou, 2025), we do not include them in our comparison.

We identify for each neural network a ZCP, an IPM, and a CP using the calibration data and considering $n_\mathrm{out} = 0, ..., 5$ outliers. For each predictor s, we compute

- its coverage $\hat{\eta}_s$ over the test set $\mathcal{M}_\mathrm{test}$, i.e, the percentage of test data points that are contained in the respective prediction sets:

$$\hat{\eta}_s = \frac{1}{n_\mathrm{test}} \sum_{(x,y)\in\mathcal{M}_\mathrm{test}} \mathbb{1}\{y \in \mathcal{Y}_s(x)\},$$

  where n_test is the number of data points in $\mathcal{M}_\mathrm{test}$, and

- its conservatism $\hat{c}_s$ over the test set $\mathcal{M}_\mathrm{test}$, which is evaluated using the average volume of the prediction sets for $n_\mathrm{y} \leq 5$:

$$\hat{c}_s = \frac{1}{n_\mathrm{test}} \sum_{(x,y)\in\mathcal{M}_\mathrm{test}} V(\mathcal{Y}_s(x)), \tag{27}$$

  where $V(\mathcal{Y}_s(x))$ computes the volume of the set $\mathcal{Y}_s(x)$. Since the computation of the exact volume becomes computationally expensive for high-dimensional zonotopes, we evaluate the conservatism of a predictor for $n_\mathrm{y} > 5$ with

$$\hat{c}_s = \frac{1}{n_\mathrm{test}} \sum_{(x,y)\in\mathcal{M}_\mathrm{test}} \sum_{i=0}^{\lceil n_\mathrm{y}/3 \rceil - 1} V_i(\mathcal{Y}_s(x)),$$

Task	$n_{\text{out}} = 0$		$n_{\text{out}} = 1$		$n_{\text{out}} = 5$	
	CP	IPM/ZCP	CP	IPM/ZCP	CP	IPM/ZCP
Energy	0.0496	0.2508	0.1048	0.3291	0.2622	0.4928
SCM1d	0.0068	0.0228	0.0183	0.0309	0.0481	0.0495
SCM20d	0.0074	0.0249	0.0200	0.0337	0.0524	0.0540
RF1	0.0130	0.0299	0.0421	0.0397	0.1088	0.0619
Photovoltaic	0.0077	0.0581	0.0207	0.0729	0.0544	0.1040
Others	0.0039	0.0201	0.0083	0.0274	0.0216	0.0447

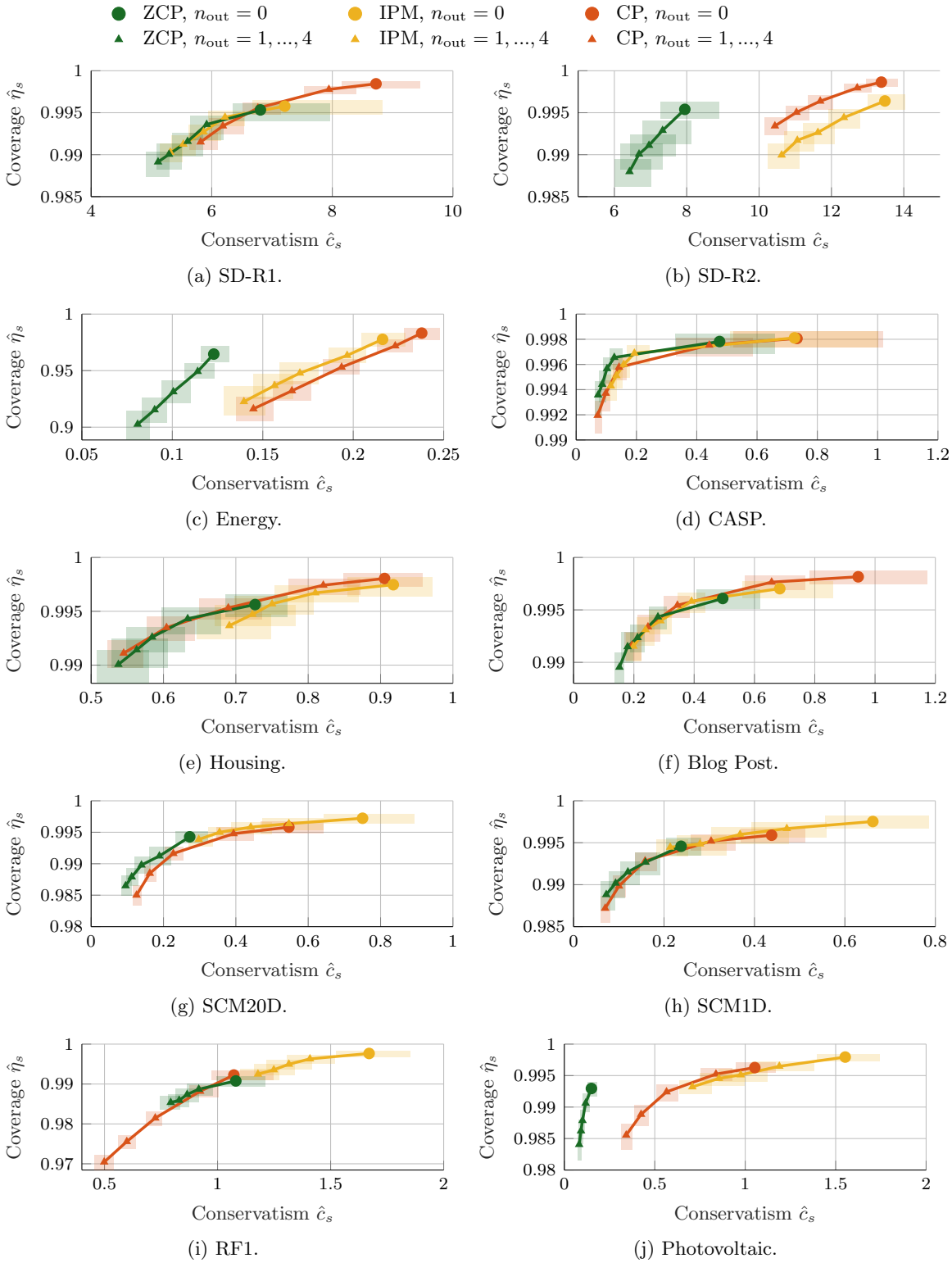
Table 2: Smallest values of ϵ for which the coverage guarantee $\mathbb{P}\{\eta(\mathcal{M}_{\text{cal}}) \geq 1 - \epsilon\} \geq 0.9$ is satisfied, across different predictors and prediction tasks. The guarantees for the regression tasks *SD-R1*, *SD-R2*, *CASP*, *Housing*, and *BlogPost* are identical and given in the last row.

where $V_i(\mathcal{Y}_s(x))$ computes the volume of the set $\mathcal{Y}_s(x)$ projected onto the dimensions $\{3i + 1, 3i + 2, 3i + 3\}$.

For each data set and predictor, we report the average coverage and conservatism across 30 trained networks, along with their 95% bootstrap confidence intervals, as shown in Fig. 2. Example prediction sets for two test data points are illustrated in Fig. 3.

Overall, ZCPs produce smaller prediction sets compared to IPMs and CPs. This is largely due to the greater flexibility of zonotopes, which can represent not only axis-aligned intervals but also more complex geometries in the output space. ZCPs offer the greatest advantage over CPs and IPMs when the output variables are correlated, as multi-dimensional intervals cannot adequately capture such dependencies. For instance, for the *SD-R2*, *Energy*, and *Photovoltaic* data sets, ZCPs produce significantly smaller prediction sets compared to CPs and IPMs, as illustrated in Fig. 2b, Fig. 2c, and Fig. 2j. This indicates that, in these cases, ZCPs are able to effectively capture and leverage output correlations, leading to less conservative uncertainty quantification. In contrast, for the *SD-R1* and *BlogPost* data sets, the prediction sets produced by ZCPs closely resemble axis-aligned intervals (see Figs. 3a and 3f), suggesting that the selected uncertainties fail to capture the inherent correlations. Consequently, as shown in Fig. 2a and Fig. 2f, ZCPs offer little reduction in conservatism for these data sets.

The increased modeling flexibility of ZCPs comes at the cost of reduced coverage guarantees on unseen data compared to CPs due to a higher risk of overfitting to the calibration set. This can also be observed in Tab. 2, which reports the smallest ϵ values satisfying (8) for a given confidence level. Since we apply CP dimension-wise—identifying one parameter and discarding n_{out} outliers for each dimension—we set $n_\theta = n_{\text{y}}$ and allow up to $n_{\text{out}} n_{\text{y}}$ outliers in (8). Despite this increased allowance for outliers, CPs achieve stronger coverage guarantees, as ZCPs and IPMs typically identify a larger number of parameters ($n_\theta > n_{\text{y}}$). The particularly low coverage observed for the *Energy* data set is attributed to its limited calibration set size.



(a) SD-R1. (b) SD-R2.

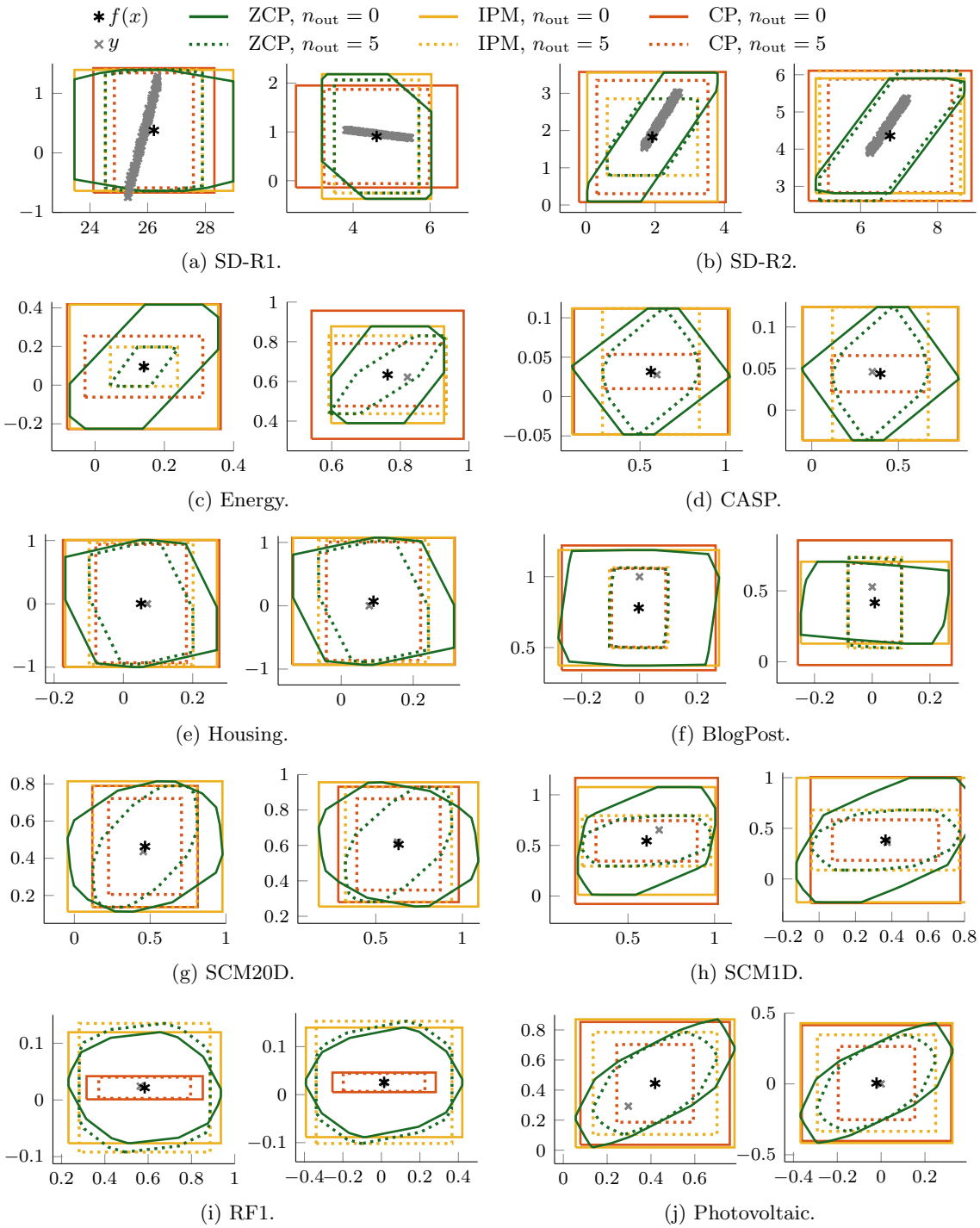
(c) Energy. (d) CASP.

(e) Housing. (f) Blog Post.

(g) SCM20D. (h) SCM1D.

(i) RF1. (j) Photovoltaic.

Figure 2: Trade-off between coverage and conservatism for different regression tasks.



(a) SD-R1. (b) SD-R2.

(c) Energy. (d) CASP.

(e) Housing. (f) BlogPost.

(g) SCM20D. (h) SCM1D.

(i) RF1. (j) Photovoltaic.

Figure 3: Prediction sets for different regression tasks projected onto the $y_1 - y_2$ plane for example data points from the test set. The true outputs are denoted by gray crosses. For the synthetic data sets SD-R1 and SD-R2, we included 100 outputs, which would have been possible for the given input using different uncertainty realizations.

6.2 Classification Tasks

For classification tasks, where a set of classes is returned if the uncertainty is high, we compare ZCPs with

- IPMs that are identified using the presented ZCP framework for classification tasks but represent the uncertainties and prediction sets with multi-dimensional intervals, and
- CPs that use the classification score function in (6).

We identify for each neural network ZCPs, IPMs, and CPs using the calibration data and considering $n_{\text{out}} = 0, ..., 5$ outliers. Each predictor s is evaluated using

- its coverage $\hat{\eta}_s$ over the test set $\mathcal{M}_{\text{test}}$, i.e, the percentage of test data points where the corresponding prediction set contains the correct class:

$$\hat{\eta}_s = \frac{1}{n_{\text{test}}} \sum_{(x,y)\in\mathcal{M}_{\text{test}}} \mathbb{1}\Big\{\text{classes}(y) \subseteq \text{classes}\big(\mathcal{Y}_s(x)\big)\Big\},$$

  where n_{test} is the number of data points in $\mathcal{M}_{\text{test}}$, and

- its conservatism $\hat{c}_s$ over the test set $\mathcal{M}_{\text{test}}$, which is evaluated using the average number of predicted classes:

$$\hat{c}_s = \frac{1}{n_{\text{test}}} \sum_{(x,y)\in\mathcal{M}_{\text{test}}} V\Big(\text{classes}\big(\mathcal{Y}_s(x)\big)\Big),$$

  where $V\Big(\text{classes}\big(\mathcal{Y}_s(x)\big)\Big)$ returns the number of elements in $\text{classes}\big(\mathcal{Y}_s(x)\big)$.

For each data set and predictor, we plot the coverage over the conservatism averaged over all 30 networks as well as their 95% bootstrap confidence intervals in Fig. 4. Similar to the regression setting, ZCPs produce less conservative prediction sets than both IPMs and CPs in the classification setting. This benefit is particularly evident for the *MNIST* and *Covertype* data sets, where ZCPs predict substantially fewer classes while allowing the same number of outliers. However, this reduction in conservatism comes at the cost of a slight decrease in coverage.

Furthermore, Fig. 5 shows representative prediction sets for IPMs and ZCPs on individual test samples. Since conformal prediction directly yields a set of classes without score-based uncertainty regions, the results of CPs are omitted from this figure. For each input x in Fig. 5, IPMs and ZCPs generate prediction sets $\mathcal{Y}(x)$, which intersect the gray area, indicating there exists a score vector $y \in \mathcal{Y}(x)$ consistent with correct classification. For the synthetic data sets *SD-C1* and *SD-C2*, both IPMs and ZCPs often produce compact prediction sets fully contained within the gray area (see left columns of Figs. 5a and 5b), reflecting high classification confidence. Furthermore, in the right columns of Figs. 5a and 5d, we see that although the deterministic prediction model $f(x)$ alone would misclassify the sample (as indicated by the black asterix lying outside the gray region), both IPMs and ZCPs correctly account for the uncertainty by generating sets that intersect the

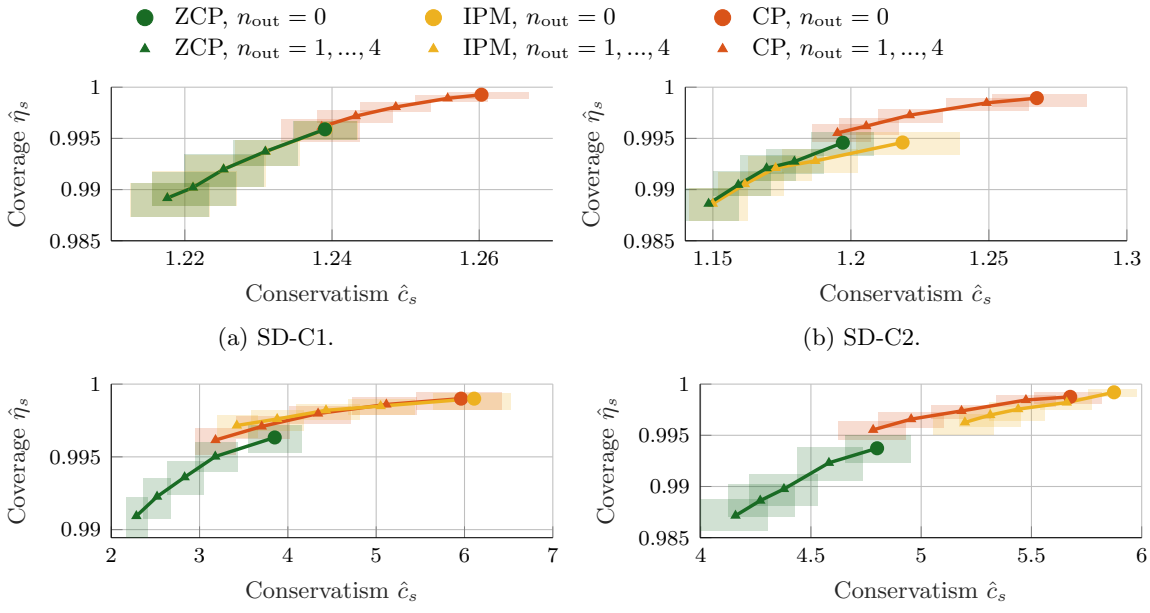
gray region—signaling the potential for correct classification. The advantage of zonotopes becomes apparent in the right column of Fig. 5c: while IPMs generate sets that extend into the white region—suggesting ambiguity—the ZCP with $n_{\text{out}} = 5$ results in a prediction set that remains strictly within the gray region, thus, excluding the incorrect class.

7 Conclusions

We have introduced zono-conformal predictors—a novel framework for uncertainty quantification that builds methodologically on interval predictor models, while aligning with conformal prediction in its practical goal of equipping arbitrary predictors with reliable uncertainty sets. Unlike conventional approaches that rely on axis-aligned intervals or separate data sets for the uncertainty modeling and calibration steps, we construct prediction zonotopes by identifying uncertainties directly within the predictor architecture. This allows the uncertainty modeling and calibration to be unified into a single, data-efficient linear program, while preserving valid coverage guarantees for unseen data. We also extended ZCPs to classification tasks and developed strategies for outlier detection that further reduce conservatism by identifying and removing anomalous data points during calibration. Our evaluation on different data sets demonstrates that ZCPs consistently yield smaller prediction sets than CPs and IPMs, particularly in multi-output settings with correlated outputs—scenarios where conventional interval-based methods struggle to capture dependencies effectively. Beyond test-time performance, ZCPs offer important practical benefits in application domains where calibration data are reused, such as in safety-critical learning-based systems. Since ZCPs maintain the same coverage over calibration data as IPMs and CPs but generate significantly smaller prediction sets, they allow for tighter safety margins without compromising safety. Nevertheless, ZCPs also come with certain limitations:

- ZCPs are calibrated by solving an optimization problem, resulting in increased computational cost compared to the calibration of standard CPs.
- The guaranteed coverage on unseen data degrades as the number of identified uncertainties increases due to potential overfitting.
- Zonotopes are restricted to convex and centrally symmetric shapes. Although our framework naturally generalizes to other set representations, a linear-programming-based construction requires that both a proxy for set volume and the containment condition $y \in \mathcal{Y}(x)$ remain linear in the parameters θ.

Future work will focus on mitigating these limitations. In particular, we plan to develop training strategies for the base predictor that promote a small set of uncertainties at user-defined locations, thereby reducing computational overhead in the calibration step and improving coverage guarantees. In parallel, exploring alternative set representations and parameterizations—capable of expressing multi-modal or non-convex uncertainties beyond a single zonotope—offers a promising avenue for enhancing the applicability of the ZCP framework. Overall, zono-conformal prediction provides a flexible, scalable, and principled framework for reliable uncertainty quantification, offering a strong foundation for safer and more effective deployment of machine learning models in real-world systems.



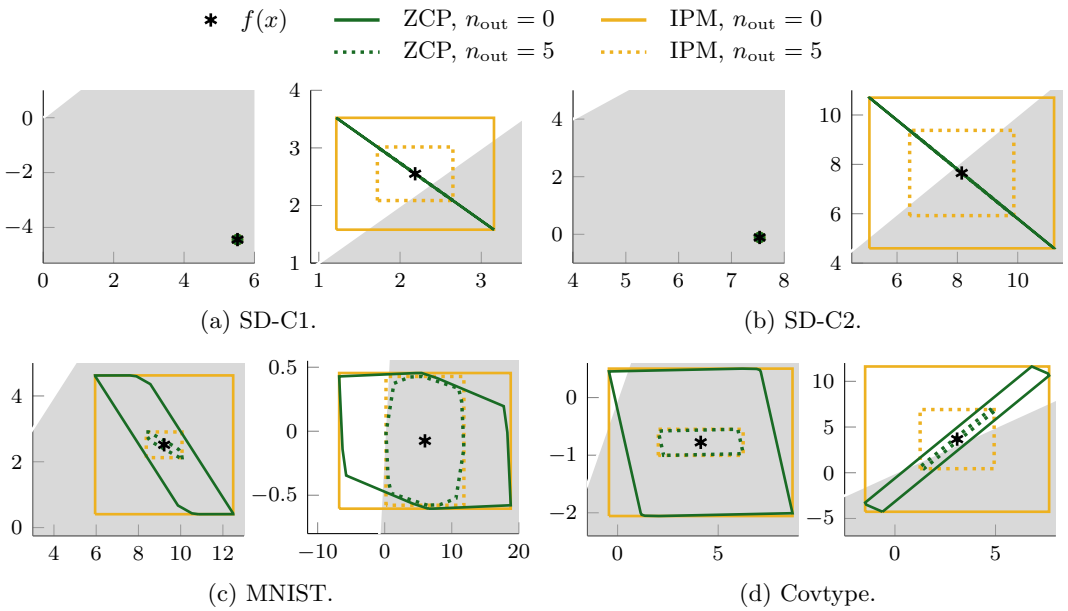
(a) SD-C1.

(b) SD-C2.

(c) MNIST.

(d) Covertype.

Figure 4: Trade-off between coverage and conservatism for different classification tasks.



(a) SD-C1.

(b) SD-C2.

(c) MNIST.

(d) Covtype.

Figure 5: Prediction sets for classification tasks for example data points from the test set. The x-axis represents the output score for the true class, while the y-axis corresponds to the output for an arbitrarily chosen incorrect class. The gray shaded region marks the domain where the true class has a higher score than the incorrect class.

Acknowledgments and Disclosure of Funding

The authors would like to thank the anonymous reviewers and the action editor for their constructive comments. The authors also want to thank Jan Jakub Kamiński for his valuable feedback on the manuscript. This work was funded by the German Research Foundation—SFB 1608 under grant number 501798263 and grant number 458030766—and supported by a fellowship within the IFI programme of the German Academic Exchange Service.

Appendix A. Data Sets

For regression tasks, we consider two synthetic and eight real-world data sets, as listed in Tab. 1. All data points of the real-world data sets are normalized to lie between 0 and 1. The synthetic and the *Photovoltaic* data sets are described in the following:

- The synthetic data set *SD-R1* contains data generated by the test function

$$g(x,u) = \begin{bmatrix} 5\sin(x_1) + x_2^2 + x_1 u_1 \\ \frac{1}{x_1^2+1} + \cos(x_2) + x_2 u_2 \end{bmatrix},$$

  where x is randomly sampled from the multi-dimensional interval $[-5 \cdot \mathbf{1},\ 5 \cdot \mathbf{1}]$ and u is sampled from the zonotope $\langle \mathbf{0},\ [0.2 \cdot \mathbf{1} \quad 0.02 \cdot v_1] \rangle$, where $v_1 = [-0.209 \quad 1.129]^\top$ was randomly sampled from a standard normal distribution.

- The synthetic data set *SD-R2* contains data generated by the test function (Sadeghi et al., 2019)

$$g(x,u) = \begin{bmatrix} 3x_1^3 + \exp(\cos(10x_2)\cos^2(5x_1)) \\ 2x_1^2 + \exp(\cos(10x_1)\cos^2(5x_2)) \end{bmatrix} + \begin{bmatrix} \exp(\sin(7.5x_3)) + u_1 \\ \exp(\sin(7.5x_3^2)) + 1.5u_2 \end{bmatrix},$$

  where x is randomly sampled from the multi-dimensional interval $[\mathbf{0},\ \mathbf{1}]$ and u is sampled from the zonotope $\langle 0.5 \cdot \mathbf{1},\ [0.5 \cdot \mathbf{1} \quad 0.05 \cdot v_2] \rangle$, where $v_2 = [0.747 \quad -0.247]^\top$ was randomly sampled from a standard normal distribution.

- The *Photovoltaic* data set is based on the hourly power generation profile of a residential photovoltaic installation from the SimBench data set *1-LV-rural2–1-sw* (Meinecke et al., 2020). The predictor is trained to predict the next four hours based on the past 48 hours. This is a common task in energy management and serves as an example where the output features are highly correlated.

For classification tasks, we consider two synthetic and two real-world data sets, as listed in Tab. 1. The two synthetic data sets are generated as follows, where the test functions are designed to have intersections, i.e., certain inputs x could have been generated by multiple test functions:

- We generate the synthetic data set *SD-C1* from the test functions

$$\begin{aligned} &\text{class 1: } x_2 = 3\sin(x_1) + u_1, \\ &\text{class 2: } x_2 = x_1^2 + u_1, \\ &\text{class 3: } x_2 = 2x_1 - 10 + u_1, \end{aligned}$$

  where x_1 and u_1 of each class are randomly sampled from the intervals $[-5,\ 5]$ and $[-2,\ 2]$, respectively.

- We generate the synthetic data set *SD-C2* from the test functions

$$\begin{aligned}
&\text{class 1: } x_3 = x_2 \sin(x_1) + u_1,\\
&\text{class 2: } x_3 = x_1^2 + x_2 + 2u_1,\\
&\text{class 3: } x_3 = 2x_1 - 10 + x_1 x_2 + 0.5u_1^2,\\
&\text{class 4: } x_3 = 2x_1 - 16 + x_2 u_1,
\end{aligned}$$

  where x_1, x_2, and u_1 of each class are randomly sampled from the intervals $[-5,\ 5]$, $[-5,\ 5]$, and $[-1,\ 1]$, respectively.

Appendix B. Ablation Studies

In this section, we evaluate the impact of different uncertainty placement strategies, cost functions, and outlier detection methods within the zono-conformal prediction framework. To manage computational complexity, we limit the calibration data sets to a maximum of 300 data points. Unless stated otherwise, ablation results are reported for the regression data sets *SD-R1*, *SD-R2*, *Energy*, and *SCM20D*. However, we observed similar qualitative trends for uncertainty placement and outlier detection across classification data sets as well. To enable consistent comparisons across different data sets and methods, we introduce the following normalized evaluation metrics:

- The normalized conservatism of a predictor s with respect to a baseline predictor $\tilde{s}$ over the calibration data can be computed with

$$\hat{c}_{s/\tilde{s}} = \frac{1}{n_{\mathrm{m}}} \sum_{(x,y)\in\mathcal{M}_{\mathrm{cal}}} \frac{V(\mathcal{Y}_s(x))}{V(\mathcal{Y}_{\tilde{s}}(x))}, \tag{28}$$

  where $V(\mathcal{Y}_s(x))$ denotes the volume of the prediction set $\mathcal{Y}_s(x)$.

- The normalized computation time of a predictor s with respect to a baseline predictor $\tilde{s}$ is defined as

$$T_{s/\tilde{s}} = \frac{T_s(\mathcal{M}_{\mathrm{cal}})}{T_{\tilde{s}}(\mathcal{M}_{\mathrm{cal}})},$$

  where $T_s(\mathcal{M}_{\mathrm{cal}})$ is the total time required for constructing predictor s from the calibration data set $\mathcal{M}_{\mathrm{cal}}$.

In all figures, methods highlighted in green represent those we consider most practical based on the trade-off between performance and efficiency. These are the default choices adopted throughout the remainder of this work. Methods shown in purple correspond to naïve baselines or existing approaches from the literature, while those in shades of blue denote other novel strategies introduced in this paper.

B.1 Uncertainty Placement Strategies

We compare several strategies for selecting a subset of uncertainties $u \in \mathbb{R}^{n_{\mathrm{u}}}$, $n_{\mathrm{u}} = \lfloor p_{\mathrm{p}} n_{\mathrm{p}} \rceil + n_{\mathrm{y}}$, from the candidate uncertainties $\tilde{u} = [u_{\mathrm{p}}^\top\ u_{\mathrm{y}}^\top]^\top \in \mathbb{R}^{n_{\mathrm{p}}+n_{\mathrm{y}}}$, with the parameter $p_{\mathrm{p}} \in [0, 1]$.

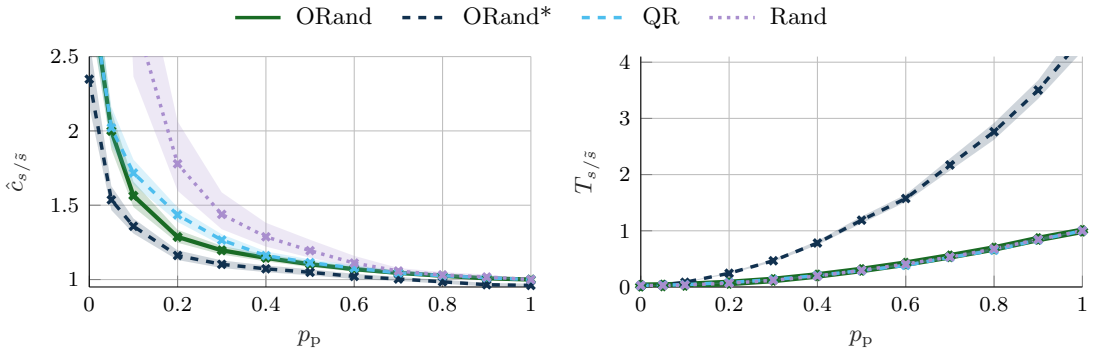
The parametric uncertainties u_p are added to all biases of the hidden layers of the neural network. Unless stated otherwise, we use the identity matrix as the generator template G_u, which yields $n_\theta = n_\mathrm{u}$ parameters to be identified. We consider the following strategies:

- *ORand*: We select all output uncertainties u_y and a random selection of $\lfloor p_\mathrm{p} n_\mathrm{p} \rceil$ parametric uncertainties, as proposed in Sec. 4.2.
- *ORand**: We select the uncertainties as in *ORand* (see Sec. 4.2), but create the generator template G_u from a horizontal concatenation of the identity matrix and n_u random generators, i.e., $n_\theta = 2n_\mathrm{u}$.
- *QR*: We select n_u uncertainties using the following deterministic QR-factorization strategy, which aims to select uncertainties whose influence is as different as possible up to some linear scaling factor.

  1. For each candidate uncertainty $\tilde{u}_{(i)}$, we compute the partial derivative $\frac{\partial \tilde{f}(x,\tilde{u})}{\partial \tilde{u}_{(i)}}|_{\tilde{u}=\mathbf{0}}$, where the partial derivative is evaluated at $\tilde{u} = \mathbf{0}$ as we assume the uncertainties are small and zero-centered. We construct the matrix V by concatenating the derivatives for the inputs $x^{(m)}$, $m = 1, \ldots, n_\mathrm{m}$, and uncertainties $\tilde{u}_{(i)}$, where the i-th column of V is given by
  $$V_{(\cdot,i)} = \mathrm{vert}_{m \in \{1,\ldots,n_\mathrm{m}\}} \left(\frac{\partial f(x^{(m)}, \tilde{u})}{\partial \tilde{u}_{(i)}}|_{\tilde{u}=\mathbf{0}} \right)$$
  and describes the general influence of uncertainty $\tilde{u}_{(i)}$ on the output.
  2. We apply the QR-factorization algorithm with column pivoting (Businger and Golub, 1965; Engler, 1997) on the matrix V to greedily select columns $V_{(\cdot,i)}$ that are as linearly independent from the previous ones as possible.
  3. We select all output uncertainties by default plus the uncertainties corresponding to the first $n_\mathrm{u} - n_\mathrm{y}$ columns of V (excluding any already-selected output uncertainties).

  Since this strategy relies on the calibration data set, the resulting coverage guarantees are weaker than those of data-independent strategies. As an alternative, a separate data set could be employed specifically for uncertainty placement.
- *Rand*: We randomly select $\lfloor p_\mathrm{p} n_\mathrm{p} \rceil + n_\mathrm{y}$ uncertainties from $\tilde{u}$.

The normalized conservatism and computation time are visualized in Fig. 6 for varying identification ratios p_p. The more uncertainties we identify by increasing p_p, the lower is the conservatism of the prediction sets but the higher is the computation time. Strategies that include all output uncertainties by default—*ORand*, *ORand**, and *QR*—consistently yield less conservative prediction sets compared to the fully random strategy *Rand*. Interestingly, the simple *ORand* approach leads to even smaller prediction sets than the more complex *QR* method. Adding extra random generators in the generator template G_u, as done in *ORand**, slightly reduces conservatism but increases computational time. Overall, the *ORand* strategy offers the best trade-off between low conservatism and computational efficiency. Since selecting too many uncertainties can negatively impact the coverage guarantees (see Sec. 5), we use $p_\mathrm{p} = 0.1$ for all other experiments.



(a) Normalized conservatism.

(b) Normalized computation time.

Figure 6: Comparison of uncertainty placement strategies. The values are normalized by the results of the predictor $\tilde{s}$, identified using *ORand* with $p_\mathrm{p} = 1$, and averaged over all networks and data sets. The shaded regions represent 95% bootstrap confidence intervals.

B.2 Cost Functions for Regression

In regression tasks, our goal is to identify uncertainties that both satisfy the conformance constraints and minimize the volume of the resulting prediction sets. The volume of a zonotope $\mathcal{Z} = \langle c, G \rangle$, with $G \in \mathbb{R}^{n \times \nu}$, is given by (Gover and Krikorian, 2010, Cor. 3.4)

$$\mathrm{volume}(\mathcal{Z}) = 2^n \sum_{1 \leq j_1 < \cdots < j_n \leq \nu} |\det(G_{j_1,\ldots,j_n})|, \tag{29}$$

where the summation considers all possible choices of $j_1, \ldots, j_n$ that satisfy $1 \leq j_1 < \cdots < j_n \leq \nu$, and $G_{j_1,\ldots,j_n}$ is constructed from columns $j_1, \ldots, j_n$ of G for all possible choices of $j_1, \ldots, j_n$ that satisfy $1 \leq j_1 < \cdots < j_n \leq \nu$. However, the volume of a zono-conformal prediction set is non-convex in the scaling factors α, making direct volume minimization computationally intractable. Instead, we evaluate alternative cost functions that encourage small prediction sets and remain linear in α:

- *Interval*: We minimize the interval norm of the prediction sets, which is the standard cost function in reachset-conformant identification (Liu et al., 2023; Lützow and Althoff, 2024), i.e.,

$$\mathrm{size}(\mathcal{Y}_{\mathrm{ZCP}}(x)) = \|\mathcal{Y}_{\mathrm{ZCP}}(x)\|_I. \tag{30}$$

- *Rotated Intervals (RI)*: The interval norm only motivates to minimize the extensions of the prediction sets along the unit vectors. To also reward small extensions along other directions, we propose to minimize the summed interval norms of the prediction sets that are randomly rotated. This leads to the cost function in (14). In the following experiments, we evaluate $n_\mathrm{r} = 5$, $n_\mathrm{r} = 10$, and $n_\mathrm{r} = 20$.

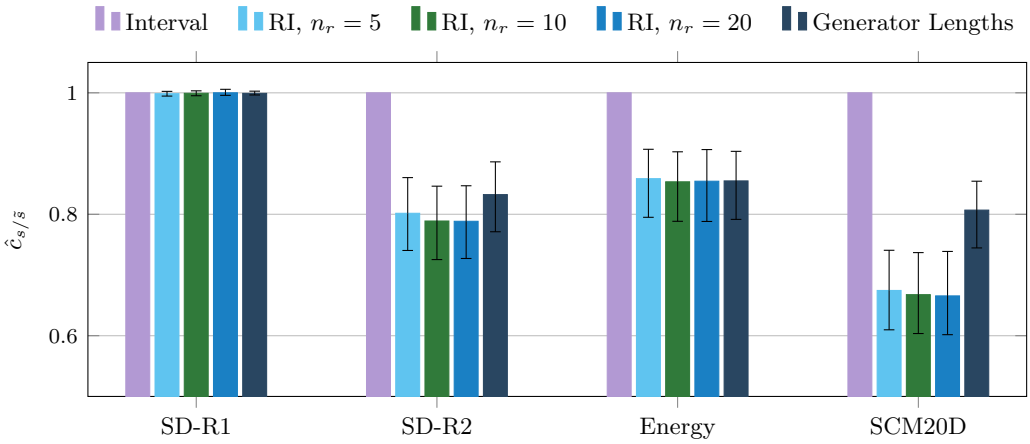


Figure 7: Comparison of cost functions for regression. The values are normalized by the results of the predictor $\tilde{s}$, identified using the *Interval* cost function, and averaged over all networks. The error bars denote the 95% bootstrap confidence intervals.

Cost Function	*SD-R1*	*SD-R2*	*Energy*	*SCM20D*
Interval	0.52	0.55	0.11	0.77
RI, $n_r = 5$	0.55	0.66	0.11	0.83
RI, $n_r = 10$	0.61	0.66	0.13	0.81
RI, $n_r = 20$	0.74	0.77	0.16	1.03
Generator Lengths	0.47	0.51	0.09	0.68

Table 3: Computation time in seconds using different cost functions.

- *Generator Lengths*: Alternatively, we can minimize the summed length of all generators of $\mathcal{Y}_{\text{ZCP}}(x) = \langle c, [g_1 \ g_2 \ \cdots \ g_{n_\theta}] \rangle$:

$$\text{size}(\mathcal{Y}_{\text{ZCP}}(x)) = \sum_{i=1}^{n_\theta} \|g_i\|_2. \tag{31}$$

The normalized conservatism for different data sets is visualized in Fig. 7, and the unnormalized computation times are reported in Tab. 3. Among all methods, the *RI* cost function consistently yields the smallest prediction sets, reducing the volume by up to a factor of 0.66 compared to the standard *Interval* cost function. While increasing the number of rotations n_r slightly improves performance, it also increases computation time due to the cost of generating random orthogonal matrices R_i. Nevertheless, since this overhead does not depend on the calibration set size n_m, the *RI* approach scales well with larger data sets. In fact, when using the full calibration set (i.e., without limiting $n_\text{m} \leq 300$), the average computation time for *RI* with $n_\text{r} = 10$ was lower than for the *Interval* cost on all data sets except the small *Energy* data set. Given the marginal improvements beyond $n_\text{r} = 10$, we adopt this value for all other experiments.

B.3 Cost Functions for Classification

In classification tasks, the objective is to minimize the number of classes predicted for each input. As smaller prediction sets reduce the likelihood of including multiple classes, we can apply the same cost functions as in the regression setting (see Sec. B.2). Additionally, we evaluate two cost functions specifically designed to penalize incorrect class scores. Specifically, we compare the following cost functions, which can all be formulated linearly in the scaling factors α:

- *Interval*: We minimize the interval norm as in (30).
- *Rotated Intervals (RI)*: We minimize the summed interval norms of randomly rotated prediction sets as in (14), where we use $n_\mathrm{r} = 10$ rotations.
- *Generator Lengths*: We minimize the summed generator lengths as in (31).
- *Score*: Alternatively, we can minimize the maximum score of incorrect classes using the cost function

$$\mathrm{size}(\mathcal{Y}_{\mathrm{ZCP}}(x)) = \sum_{i \notin \mathrm{classes}(y)} \max_{z \in \mathcal{Y}_{\mathrm{ZCP}}(x)} z_{(i)}, \tag{32}$$

  where y encodes the correct classes for the input x.
- *Score Difference*: To motivate robust correct predictions, we minimize the maximum score difference between incorrect and correct classes, i.e.,

$$\mathrm{size}(\mathcal{Y}_{\mathrm{ZCP}}(x), y) = \sum_{i \notin \mathrm{classes}(y)} \sum_{j \in \mathrm{classes}(y)} d_{ij}, \tag{33}$$

  where the score difference $d_{ij} = \max_{z \in \mathcal{Y}_{\mathrm{ZCP}}(x)} z_{(i)} - z_{(j)}$ is the maximum difference between the classification score for class i and the score for class j over the prediction set $\mathcal{Y}_{\mathrm{ZCP}}(x)$.

The normalized conservatism across the classification data sets is shown in Fig. 8. All evaluated cost functions yield similarly sized prediction sets, with minor variations depending on the data set. For methodological consistency and simplicity, we adopt the same cost function used in the regression setting—namely, *RI* with $n_r = 10$—for all classification experiments.

B.4 Outlier Detection

We compare the following outlier detection methods, each aiming to identify and remove n_out measurements from the calibration data set that most effectively reduce the identification cost:

- *Search*: the exhaustive search over boundary points as described in Sec. 5.1.1,
- *SearchG*: the greedy search over boundary points as described in Sec. 5.1.2,
- *MILP*: the mixed-integer linear programs as described in Sec. 5.1.3,

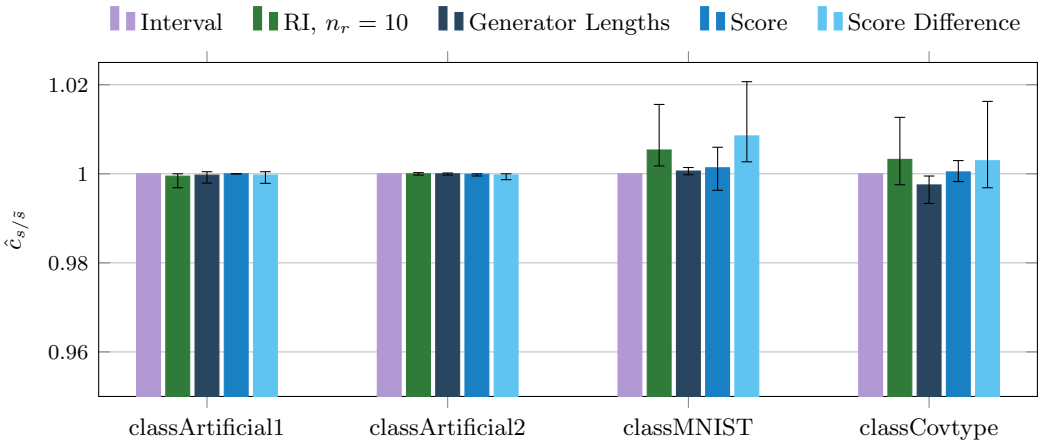


Figure 8: Comparison of cost functions for classification. The values are normalized by the results of the predictor $\tilde{s}$, identified using the *Interval* cost function, and averaged over all networks. The error bars denote the 95% bootstrap confidence intervals.

- *RMSE*: a heuristic method that removes the n_{out} data points with the largest root-mean-square prediction error $\sqrt{e^{(m)\top} e^{(m)}}$, with $e^{(m)} = y^{(m)} - f(x^{(m)})$.

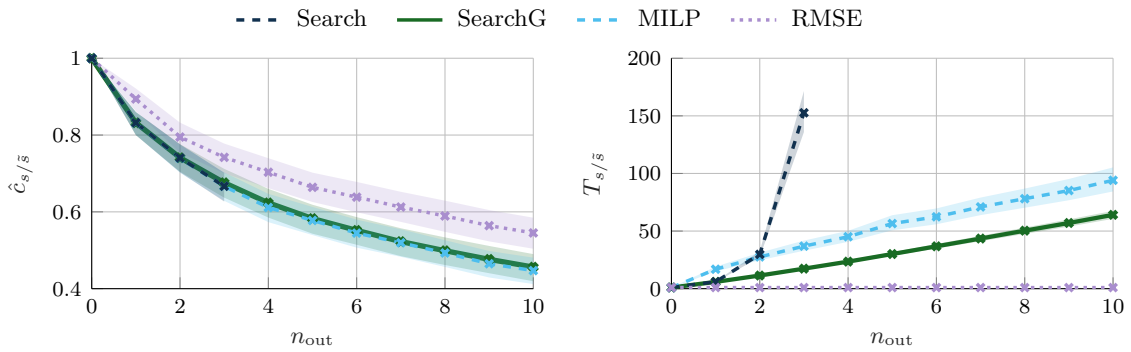
Fig. 9 shows the normalized conservatism and the normalized computation time for different numbers of removed outliers n_{out}. The exhaustive search strategy *Search* leads to the optimal identification cost but is computationally feasible only for small n_{out}. The greedy variant *SearchG* substantially reduces computation time while maintaining near-optimal identification performance. The *MILP* method consistently reaches optimal identification costs and is more efficient than *Search*, though it leads to slightly higher computation times than *SearchG*. In contrast, the *RMSE*-based approach is the most computationally efficient but results in noticeably higher identification costs. Given its favorable trade-off between accuracy and efficiency, we adopt *SearchG* as the default outlier detection method in all other experiments.

References

M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

T. Alamo, J. Bravo, and E. Camacho. Guaranteed state estimation by zonotopes. In *IEEE Conference on Decision and Control*, volume 6, pages 5831–5836, 2003.

M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *ACM International Conference on Hybrid Systems: Computation and Control*, pages 173–182, 2013.



(a) Normalized Conservatism.

(b) Normalized Computation Time.

Figure 9: Comparison of outlier detection methods. The values are normalized by the results of the predictor $\tilde{s}$, which is identified without removing any measurements, i.e., $n_{\text{out}} = 0$, and averaged over all networks and data sets. The shaded regions represent 95% bootstrap confidence intervals.

M. Althoff. Checking and establishing reachset conformance in CORA 2023. In *Workshop on Applied Verification of Continuous and Hybrid Systems*, pages 9–33, 2023.

M. Althoff and J. M. Dolan. Reachability computation of low-order models for the safety verification of high-order road vehicle models. In *American Control Conference*, pages 3559–3566, 2012.

M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of linear systems with uncertain parameters and inputs. In *IEEE Conference on Decision and Control*, pages 726–732, 2007.

A. N. Angelopoulos and S. Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

J. A. Blackard and D. J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, 1999.

J. Blesa, V. Puig, and J. Saludes. Identification for passive robust fault detection using zonotope-based set-membership approaches. *International Journal of Adaptive Control and Signal Processing*, 25(9):788–812, 2011.

S. Braun, L. Aolaritei, M. I. Jordan, and F. Bach. Minimum volume conformal sets for multivariate regression. *Preprint*, 2025. https://arxiv.org/abs/2503.19068.

P. Businger and G. H. Golub. Linear least squares solutions by Householder transformations. *Numerische Mathematik*, 7(6):269–276, 1965.

G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

M. C. Campi, G. C. Calafiore, and S. Garatti. Interval predictor models: Identification and reliability. *Automatica*, 45(2):382–392, 2009.

M. C. Campi, S. Garatti, and F. A. Ramponi. Non-convex scenario optimization with application to system identification. In *IEEE Conference on Decision and Control*, pages 4023–4028, 2015.

O. Cartagena, S. Parra, D. Muñoz-Carpintero, L. G. Marín, and D. Sáez. Review on fuzzy and neural prediction interval modelling for nonlinear dynamical systems. *IEEE Access*, 9:23357–23384, 2021.

S. B. Chabane, C. S. Maniu, T. Alamo, E. Camacho, and D. Dumur. Improved set-membership estimation approach based on zonotopes and ellipsoids. In *European Control Conference*, pages 993–998, 2014.

A. Cini, A. Jenkins, D. Mandic, C. Alippi, and F. M. Bianchi. Relational conformal prediction for correlated time series. In *International Conference on Machine Learning*, 2025.

C. Combastel. Zonotopes and Kalman observers: Gain optimality under distinct uncertainty paradigms and robust convergence. *Automatica*, 55:265–273, 2015.

R. Coppola, A. Peruffo, L. Lindemann, and M. Mazo. Scenario approach and conformal prediction for verification of unknown systems via data-driven abstractions. In *European Control Conference*, pages 558–563, 2024.

L. G. Crespo, S. P. Kenny, B. K. Colbert, and T. Slagel. Interval predictor models for robust system identification. In *IEEE Conference on Decision and Control*, pages 872–879, 2021.

M. de Angelis, R. Rocchetta, A. Gray, and S. Ferson. Constructing consonant predictive beliefs from data with scenario theory. In *International Symposium on Imprecise Probabilities and Their Applications*, 2021.

V. Dheur, M. Fontana, Y. Estievenart, N. Desobry, and S. B. Taieb. A unified comparative study with generalized conformity scores for multi-output conformal regression. In *International Conference on Machine Learning*, 2025.

H. Engler. The behavior of the QR-factorization algorithm with column pivoting. *Applied Mathematics Letters*, 10(6):7–11, 1997.

Z. Fang, A. Tan, and J. Huang. CONTRA: Conformal prediction region via normalizing flow transformation. In *International Conference on Learning Representations*, 2025.

S. Feldman, S. Bates, and Y. Romano. Calibrated multiple-output quantile regression with representation learning. *Journal of Machine Learning Research*, 24(24):1–48, 2023.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

S. Garatti, M. C. Campi, and A. Carè. On a class of interval predictor models with universal reliability. *Automatica*, 110(C), 2019. article no. 108542.

A. Girard. Reachability of uncertain linear systems using zonotopes. In *ACM International Conference on Hybrid Systems: Computation and Control*, pages 291–305, 2005.

E. Goan and C. Fookes. Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science*, pages 45–87, 2020.

E. Gover and N. Krikorian. Determinants and the volumes of parallelotopes and zonotopes. *Linear Algebra and its Applications*, 433(1):28–40, 2010.

A. Gray, V. Gopakumar, S. Rousseau, and S. Destercke. Guaranteed prediction sets for functional surrogate models. In *Conference on Uncertainty in Artificial Intelligence*, 2025.

F. Gruber and M. Althoff. Scalable robust safety filter with unknown disturbance set. *IEEE Transactions on Automatic Control*, 68(12):7756–7770, 2023.

L. Huang, B.-L. Zhang, and Q. Huang. Robust interval regression analysis using neural networks. *Fuzzy Sets and Systems*, 97(3):337–347, 1998.

H. Ishibuchi and H. Tanaka. Several formulations of interval regression analysis. In *Sino-Japan Joint Meeting on Fuzzy Sets and Systems*, pages 173–182, 1990.

H. M. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi. Neural network-based uncertainty quantification: A survey of methodologies and applications. *IEEE Access*, 6: 36218–36234, 2018.

A. Khosravi, S. Nahavandi, D. Srinivasan, and R. Khosravi. Constructing optimal prediction intervals by using neural networks and bootstrap method. *IEEE Transactions on Neural Networks and Learning Systems*, 26(8):1810–1815, 2015.

W. Kühn. Rigorously computed orbits of dynamical systems without the wrapping effect. *Computing*, 61(1):47–67, 1998.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *International Conference on Neural Information Processing Systems*, pages 6405–6416, 2017.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

J. Lei, J. Robins, and L. W. and. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013.

A. Lin and S. Bansal. Verification of neural reachable tubes via scenario optimization and conformal prediction. In A. Abate, M. Cannon, K. Margellos, and A. Papachristodoulou, editors, *Conference on Learning for Dynamics and Control*, volume 242, pages 719–731, 2024.

S. B. Liu, B. Schürmann, and M. Althoff. Guarantees for real robotic systems: Unifying formal controller synthesis and reachset-conformant identification. *IEEE Transactions on Robotics*, 39(5):3776–3790, 2023.

R. Luo and Z. Zhou. Volume-sorted prediction set: Efficient conformal prediction for multi-target regression. *Preprint*, 2025. https://arxiv.org/abs/2503.02205.

L. Lützow and M. Althoff. Scalable reachset-conformant identification of linear systems. *IEEE Control Systems Letters*, 8:520–525, 2024.

L. Lützow and M. Althoff. Reachset-conformant system identification. *IEEE Transactions on Automatic Control (Early Access)*, pages 1–16, 2025a.

L. Lützow and M. Althoff. Recursive identification of reachset-conformant models using constraint underapproximation. In *American Control Conference*, pages 2530–2537, 2025b.

S. Meinecke, D. Sarajlić, S. R. Drauz, A. Klettke, L.-P. Lauven, C. Rehtanz, A. Moser, and M. Braun. SimBench — a benchmark dataset of electric power systems to compare innovative solutions based on power flow analysis. *Energies*, 13(12), 2020. article no. 3290.

S. Messoudi, S. Destercke, and S. Rousseau. Copula-based conformal prediction for multi-target regression. *Pattern Recognition*, 120, 2021. article no. 108101.

S. Messoudi, S. Destercke, and S. Rousseau. Ellipsoidal conformal inference for multi-target regression. In *Symposium on Conformal and Probabilistic Prediction with Applications*, volume 179, pages 294–306, 2022.

N. O'Sullivan, L. Romao, and K. Margellos. Bridging conformal prediction and scenario optimization. In *IEEE Conference on Decision and Control*, pages 6114–6121, 2025.

R. Rocchetta, Q. Gao, and M. Petkovic. Soft-constrained interval predictor models and epistemic reliability intervals: A new tool for uncertainty quantification with limited experimental data. *Mechanical Systems and Signal Processing*, 161:107973, 2021.

R. Rocchetta, A. Mey, and F. A. Oliehoek. A survey on scenario theory, complexity, and compression-based learning and generalization. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):16985–16999, 2024.

H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff. Reachset conformance testing of hybrid automata. In *ACM International Conference on Hybrid Systems: Computation and Control*, pages 277–286, 2016.

H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff. Model conformance for cyber-physical systems: A survey. *ACM Transactions on Cyber-Physical Systems*, 3(3), 2019. article no. 30.

J. Sadeghi, M. de Angelis, and E. Patelli. Efficient training of interval neural networks for imprecise training data. *Neural Networks*, 118:338–351, 2019.

L. Schäfer, F. Gruber, and M. Althoff. Scalable computation of robust control invariant sets of nonlinear systems. *IEEE Transactions on Automatic Control*, 69(2):755–770, 2024.

K. Stankevičiūtė, A. Alaa, and M. van der Schaar. Conformal time-series forecasting. In *International Conference on Neural Information Processing Systems*, 2021. article no. 475.

S. H. Sun and R. Yu. Copula conformal prediction for multi-step time series prediction. In *International Conference on Learning Representations*, 2024.

A. Tsanas and A. Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49: 560–567, 2012.

G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. MULAN: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

R. Tumu, M. Cleaveland, R. Mangharam, G. J. Pappas, and L. Lindemann. Multi-modal conformal prediction regions by optimizing convex shape templates. In *Conference on Learning for Dynamics and Control*, 2024.

V. Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74:9–28, 2015.

V. Vovk, A. Gammerman, and C. Saunders. Machine-learning applications of algorithmic randomness. In *International Conference on Machine Learning*, pages 444–453, 1999.

C. Xu, H. Jiang, and Y. Xie. Conformal prediction for multi-dimensional time series by ellipsoidal sets. In *International Conference on Machine Learning*, 2024. article no. 2268.