Epistemic Uncertainty in Conformal Scores: A Unified Approach

Luben M. C. Cabezas[1, 2]   **Vagner S. Santos**[1]   **Thiago R. Ramos**[1]   **Rafael Izbicki**[1]

[1]Department of Statistics, Federal University of São Carlos, São Carlos, São Paulo, Brazil
[2]Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, São Paulo, Brazil

Abstract

Conformal prediction methods create prediction bands with distribution-free guarantees but do not explicitly capture epistemic uncertainty, which can lead to overconfident predictions in data-sparse regions. Although recent conformal scores have been developed to address this limitation, they are typically designed for specific tasks, such as regression or quantile regression. Moreover, they rely on particular modeling choices for epistemic uncertainty, restricting their applicability. We introduce EPICSCORE, a model-agnostic approach that enhances any conformal score by explicitly integrating epistemic uncertainty. Leveraging Bayesian techniques such as Gaussian Processes, Monte Carlo Dropout, or Bayesian Additive Regression Trees, EPICSCORE adaptively expands predictive intervals in regions with limited data while maintaining compact intervals where data is abundant. As with any conformal method, it preserves finite-sample marginal coverage. Additionally, it also achieves asymptotic conditional coverage. Experiments demonstrate its good performance compared to existing methods. Designed for compatibility with any Bayesian model, but equipped with distribution-free guarantees, EPICSCORE provides a general-purpose framework for uncertainty quantification in prediction problems.

1 INTRODUCTION

Machine learning models traditionally focus on point predictions, estimating target variables from input features. However, understanding prediction uncertainty is crucial in many applications [Horta et al., 2015, Freeman et al., 2017, Izbicki and Lee, 2017, McAllister et al., 2017, Schmidt et al., 2020, Selvan et al., 2020, Vazquez and Facelli, 2022, Csillag et al., 2023, Mian et al., 2024, Valle et al., 2024, Fröhlich et al., 2025]. This has led to increased interest in uncertainty quantification methods, particularly conformal prediction, which constructs predictive regions with finite-sample validity under mild i.i.d. assumptions [Vovk et al., 2005, Shafer and Vovk, 2008]. Unlike probabilistic models that rely on asymptotic assumptions or priors, conformal methods provide a distribution-free framework with guaranteed coverage.

Conformal prediction works by designing a non-conformity score, $s(\mathbf{x}, y)$, which measures the degree to which a given label value y aligns with the feature values $\mathbf{x}$ of an instance. Given a new input $\mathbf{x}_{\text{new}}$, the method constructs a predictive region by inverting the non-conformity score at a given confidence level (see Section 2.1). The choice of $s(\mathbf{x}, y)$ is critical, as it directly influences the shape and informativeness of the resulting predictive regions [Angelopoulos and Bates, 2021, Izbicki, 2025]. For instance, in regression problems, a standard choice is $s(\mathbf{x}, y) = |y - g(\mathbf{x})|$, where $g(\mathbf{x})$ is a point prediction for Y, typically an estimate of the regression function $\mathbb{E}[Y|\mathbf{x}]$ [Lei et al., 2018]. Another common option is $s(\mathbf{x}, y) = \max\{\widehat{q}_{\alpha_1}(\mathbf{x}) - y, y - \widehat{q}_{\alpha_2}(\mathbf{x})\}$, where $\widehat{q}_{\alpha_1}, \widehat{q}_{\alpha_2}$ are quantile estimates of $Y|\mathbf{x}$ [Romano et al., 2019].

Despite offering distribution-free guarantees, standard conformal scores only capture *aleatoric uncertainty*, which arises from inherent randomness in the data generation process - specifically, the fact that $\mathbf{x}$ does not uniquely determine y. For example, in the cases discussed above, both $E[Y|\mathbf{x}]$ and $q_\alpha(\mathbf{x})$ reflect this form of uncertainty. However, an equally important source of uncertainty is *epistemic uncertainty*, which stems from limitations in training data and the resulting lack of knowledge about the true data-generating process [Hüllermeier and Waegeman, 2021].

Although conformal sets do take epistemic uncertainty into account [Angelopoulos and Bates, 2021] , the conformal scores themselves typically do not. It follows that conformal inference often leads to misleadingly narrow predictive intervals in regions with little or no training data. Figure 1

illustrates this issue in a regression problem (see Appendix A.1 for technical details): in the range $x \in (7, 8)$, there is essentially no training data, so we expect predictive regions to widen, reflecting the increased uncertainty. However, standard conformal methods (e.g., regression-split) instead produce overconfident, narrow intervals in this region. To address this limitation, we propose a novel approach that augments any conformal score with a measure of epistemic uncertainty. As shown in the figure, our method (EPICSCORE) successfully expands the predictive regions where data is scarce, providing a more reliable uncertainty quantification.
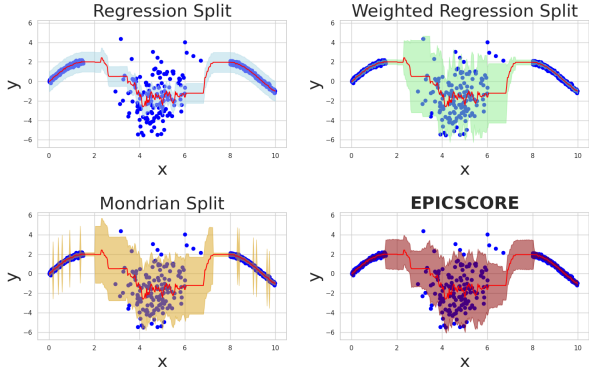

Figure 1: A comparison of predictive intervals from standard split-conformal regression and our proposed EPICSCORE approach. While all methods maintain valid marginal coverage, standard conformal prediction often produces overconfidently narrow intervals in the data-scarce region (e.g. $x \in (7, 8)$). Our method explicitly accounts for epistemic uncertainty, resulting in appropriately widened predictive intervals that better reflect total uncertainty when extrapolating beyond the training distribution.

This limitation also appears in classification tasks, where conventional conformal methods produce overconfident prediction sets for test instances outside the training distribution. Using a ResNet-34 pre-trained on ImageNet (see Appendix A.2 for details and additional results), Figure 2 compares Adaptive Prediction Sets (APS, Romano et al. 2020) and EPICSCORE on CIFAR-100 images. Both maintain valid coverage, but EPICSCORE explicitly quantifies epistemic uncertainty: prediction sets expand for outliers while remaining concise for in-distribution examples. EPICSCORE also shows improved Size-Stratified Coverage (SSC; Angelopoulos and Bates 2021) by 33% over APS, thus producing better prediction sets.

1.1 NOVELTY

We introduce EPICSCORE (Epistemic Conformal Score), a novel nonconformity score that explicitly integrates epistemic uncertainty into the conformal prediction framework. Our key innovation is modeling the epistemic uncertainty of any given conformal score $s(\mathbf{x}, y)$ using a Bayesian process. This allows us to refine uncertainty quantification, particularly in regions of the feature space where data are sparse.

EPICSCORE is flexible and can leverage various Bayesian approaches, including Gaussian Processes, Bayesian Additive Regression Trees, or even approximations such as Neural Networks with Monte Carlo Dropout. By incorporating these probabilistic models, our method dynamically adjusts predictive regions to account for both aleatoric and epistemic uncertainty.

Despite EPICSCORE using Bayesian models, it preserves the finite-sample validity guarantees of conformal prediction and also achieves asymptotic conditional coverage, a property that many existing conformal approaches lack. Furthermore, EPICSCORE can be applied on top of any existing conformal score, making it an enhancement rather than a replacement.

1.2 RELATION TO OTHER WORK

Several recent frameworks have attempted to model epistemic uncertainty in machine learning predictions (see, e.g., He et al. 2023, Tyralis and Papacharalampous 2024, Wang et al. 2025 and references therein). However, these methods generally lack guarantees on the coverage of their estimates.

In contrast, conformal prediction offers valid coverage guarantees under the relatively weak assumption of i.i.d. data. Since its introduction [Vovk et al., 2005, Shafer and Vovk, 2008], the method has advanced significantly in both theoretical foundations and practical applications [Lei et al., 2018, Angelopoulos and Bates, 2021, Fontana et al., 2023, Manokhin, 2024]. A key challenge in this area has been enhancing the quality of predictive regions, particularly by achieving conditional coverage, which ensures validity conditional on specific feature values.

Since exact conditional coverage is unattainable without strong assumptions [Lei and Wasserman, 2014], research has focused on two strategies: (1) locally tuning cutoffs to adapt to the data distribution [Boström and Johansson, 2020, Foygel Barber et al., 2021, Guan, 2023, Cabezas et al., 2024, 2025], and (2) designing conformal scores that achieve conditional coverage asymptotically [Romano et al., 2019, Izbicki et al., 2020, Chernozhukov et al., 2021, Izbicki et al., 2022, Plassier et al., 2024]. Among these, the score introduced by Dheur et al. [2025, Eq. 14] is particularly relevant to our approach. This method transforms a nonconformity score s via its estimated cumulative distribution function, $s'(\mathbf{x}; y) = \widehat{F}(s(\mathbf{x}, y)|\mathbf{x})$, improving conditional coverage. While s' ensures asymptotic conditional coverage, it does not explicitly model epistemic uncertainty. Our approach builds on this idea by incorporating epistemic
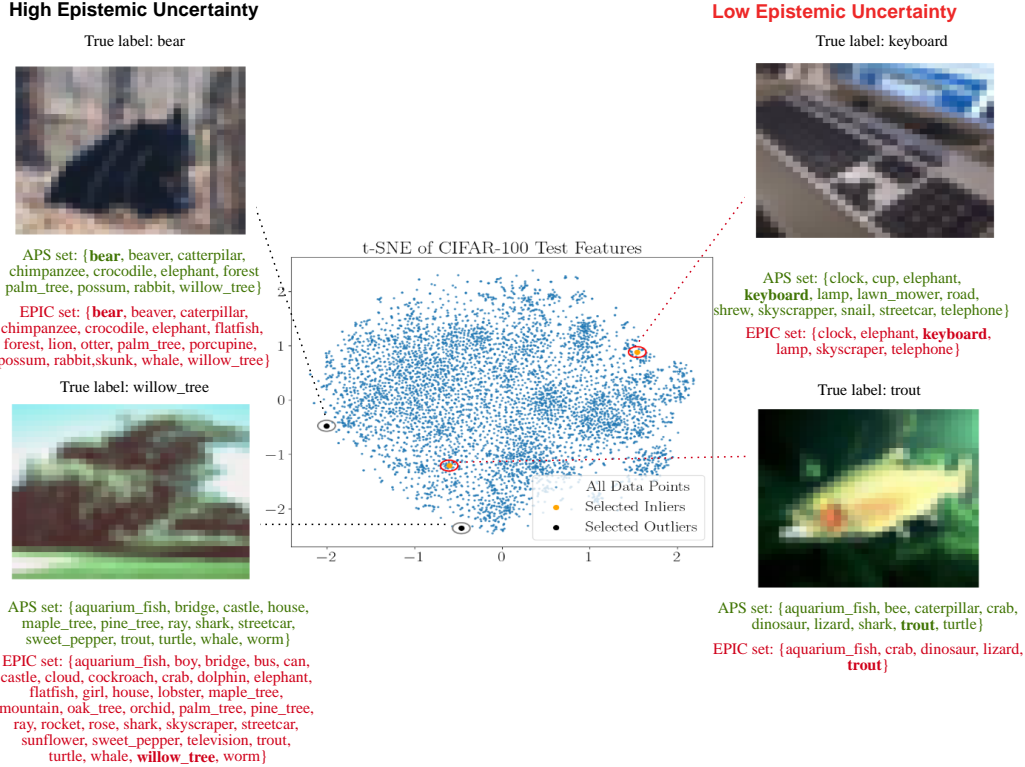
Figure 2: Prediction sets from Adaptive Prediction Sets (APS) versus the proposed EPICSCORE approach on CIFAR-100 images. Both methods maintain valid coverage, but EPICSCORE explicitly quantifies epistemic uncertainty, resulting in adaptively expanded prediction sets for outlier images (e.g., those in data-sparse regions) while remaining concise for in-distribution examples.

uncertainty in s'. In particular, we show that as the calibration sample grows, EPICSCORE achieves asymptotic conditional coverage (Theorem 2).

As in EPICSCORE, Bayesian ideas are often used in non-conformal frequentist inference, including what Wasserman [2011] terms "Frasian inference". Similar ideas appear in I. J. Good's (Good 1992 and others) advocacy for a Bayes/non-Bayes compromise, in recent work on likelihood-free frequentist inference [Masserano et al., 2023, Dalmasso et al., 2024] and Bayes-optimal sets with frequentist coverage control [Hoff, 2023]. While EPICSCORE's base model is Bayesian, the final conformal regions remain distribution-free and retain their frequentist distribution–free coverage guarantees, whichever the prior is.

Bayesian methods have also been explored as a means to incorporate epistemic uncertainty into conformal prediction. One approach is to use Bayesian predictive sets and subsequently apply conformal methods to adjust their marginal coverage. However, existing techniques typically do not leverage existing conformal scores explicitly, and do not lead to asymptotic conditional coverage [Vovk et al., 2005, Wasserman, 2011, Fong and Holmes, 2021].

Other recent studies have attempted to incorporate epistemic uncertainty directly into existing conformal scores. Cocheteux et al. [2025] modifies the Weighted regression-split nonconformity score [Lei et al., 2018], $s(\mathbf{x}, y) = |y - g(\mathbf{x})|/\sigma(\mathbf{x})$, by redefining $\sigma(\mathbf{x})$ to capture epistemic uncertainty about Y, rather than aleatoric uncertainty only as in the original formulation. This uncertainty is estimated via Monte Carlo Dropout, which we also employ in some experiments. However, our approach is more flexible, as it accommodates any nonconformity score and, unlike this method, ensures asymptotic conditional coverage.

Another approach is proposed by Rossellini et al. [2024], who modify the conformal quantile regression (CQR; Romano et al. 2019) score function, $s(\mathbf{x}, y) = \max\{\widehat{q}_{\alpha_1}(\mathbf{x}) - y, y - \widehat{q}_{\alpha_2}(\mathbf{x})\}$, to account for epistemic uncertainty in the quantile estimates. While this adaptation shows promising results, it is restricted to quantile regression models and does not generalize to other conformal scores. Moreover, it necessarily measures epistemic uncertainty using ensembles.

Other efforts to model epistemic uncertainty include Jaber et al. [2024] and Pion and Vazquez [2024], which integrate conformal methods with Gaussian processes. Although these approaches yield calibrated sets for Gaussian processes, they are tailored to this class of models and cannot

be applied to other frameworks, such as Bayesian Additive Regression Trees [Chipman et al., 2012]. In contrast, EPICSCORE is model-agnostic (in the sense that it can be used on top of any nonconformity score) and can be applied to any Bayesian model for epistemic uncertainty.

2 METHODOLOGY

2.1 REVIEW OF CONFORMAL PREDICTION

Conformal prediction constructs valid prediction regions $R(\mathbf{X}_{n+1})$ under minimal assumptions. A widely used approach is split conformal prediction [Papadopoulos et al., 2002, Lei et al., 2018], which partitions the data into a training set $\mathcal{D}_{\text{train}}$ and a calibration set $\mathcal{D}_{\text{cal}} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$. The training set is used to fit a nonconformity score $s(\mathbf{x}, y)$ such as those described in the introduction. The conformal prediction region is given by

$$R(\mathbf{x}_{n+1}) = \{y : s(\mathbf{x}_{n+1}, y) \leq t_{1-\alpha}\}.$$

The value $t_{1-\alpha}$ is set using the calibration set. Concretely, $t_{1-\alpha}$ is set to be the $(1 - \alpha)$-quantile of the calibration scores,

$$t_{1-\alpha} = \text{Quantile}_{1-\alpha}\{s(\mathbf{X}_i, Y_i) : (\mathbf{X}_i, Y_i) \in \mathcal{D}_{\text{cal}}\}.$$

This construction ensures that the prediction set for a new observation $(\mathbf{X}_{n+1}, Y_{n+1})$ satisfies marginal coverage:

$$\mathbb{P}(Y_{n+1} \in R(\mathbf{X}_{n+1})) \geq 1 - \alpha.$$

The choice of s is critical in determining the shape and other properties of R. In the next section, we introduce a new conformal score that measures the epistemic uncertainty around any given score s.

2.2 OUR APPROACH - EPICSCORE

We assume that a nonconformity score $s(\mathbf{x}, y)$ is already defined based on the training set. Our starting point is to define a family of distributions that model the aleatoric uncertainty of $s(\mathbf{X}, Y)$ given $\mathbf{X}$, which is a set of distributions indexed by a parameter $\theta \in \Theta$. We denote this family by $\mathcal{F} = \{f(s|\mathbf{x}, \theta) : \theta \in \Theta\}$. This formulation is very general; Θ may even represent a nonparametric space.

To construct EPICSCORE, we adopt a Bayesian model that places a prior distribution over Θ (or equivalently, over $\mathcal{F}$). For simplicity, we assume this prior has a density $f(\theta)$, though the method is generally applicable. The choice of prior plays a crucial role: it can be tailored to capture epistemic uncertainty in the data-generating process (via uninformative or weakly informative priors) or to introduce stronger regularization into the conformal score distribution (via more concentrated priors). In this work, we focus on

enhancing the representation of epistemic uncertainty and therefore adopt non-informative priors. However, a balance between capturing uncertainty and enforcing regularization can be explored through prior specification (see Appendix D.5 for a discussion on this trade-off).

In our experiments, we use Gaussian Processes (GP), Bayesian Additive Regression Trees (BART), and approximate Bayesian Mixture Density Networks (MDNs) with Monte Carlo dropout. However, EPICSCORE is compatible with any Bayesian modeling approach that yields a posterior predictive distribution. These three models were chosen for their complementary strengths: BART offers flexibility and interpretability, naturally handling categorical features; GPs provide expressive nonparametric inference with scalable tractability via variational approximations; and MDNs with Monte Carlo dropout are highly scalable, making them well-suited for large-scale or high-dimensional problems.

We update the prior distribution $f(\theta)$ using a subset of the calibration set $\mathcal{D}_{\text{cal}}$. Specifically, $\mathcal{D}_{\text{cal}}$ is split into two disjoint subsets: $\mathcal{D}_{\text{cal},1}$ and $\mathcal{D}_{\text{cal},2}$. The first subset, $\mathcal{D}_{\text{cal},1}$ is transformed into the dataset

$$D = \{(\mathbf{X}, S) : (\mathbf{X}, Y) \in \mathcal{D}_{\text{cal},1}, S = s(\mathbf{X}, Y)\}.$$

Using this transformed dataset, we compute the posterior distribution $f(\theta|D)$, which reflects the updated epistemic uncertainty about the data-generating process after observing this data. Our Bayesian model assumes that, given θ, the data points $(\mathbf{X}, S)$ are independent and share the same conditional distribution $f(s|\mathbf{x}, \theta)$. Then, we derive the predictive cumulative distribution

$$F(s|\mathbf{x}, D) = \int F(s|\mathbf{x}, \theta) f(\theta|D) d\theta, \qquad (1)$$

where $F(s|\mathbf{x}, \theta)$ is the CDF given by model θ.

The posterior $f(\theta|D)$ is not computed explicitly. Instead, Eq. 1 is approximated either via Monte Carlo sampling (as with BART and MC-Dropout) or through a variational approximation of $f(\theta|D)$ (as in the variational GP).

Finally, our modified nonconformity score, EPICSCORE, is defined as

$$s'(\mathbf{x}, y) = F(s(\mathbf{x}, y)|\mathbf{x}, D). \qquad (2)$$

By construction, s' incorporates epistemic uncertainty into s by averaging the original score distribution, $F(s|\mathbf{x}, \theta)$, over the posterior $f(\theta|D)$, thus propagating the uncertainty about θ throughout the model.

Once s' is computed, the prediction region for a new sample point is obtained using the standard split conformal method, with s' serving as the nonconformity score. Specifically, s' is evaluated for every sample in the second subset, $\mathcal{D}_{\text{cal},2}$. The $(1 - \alpha)$-quantile of these values, denoted as $t_{1-\alpha}$, is then used to construct the prediction region

$$R_{\text{EPIC}}(\mathbf{x}_{n+1}) = \{y : s'(\mathbf{x}_{n+1}, y) \leq t_{1-\alpha}\},$$

which, by the definition of s', can be expressed in terms of the original nonconformity score s as

$$R_{\text{EPIC}}(\mathbf{x}_{n+1}) = \{y : s(\mathbf{x}_{n+1}, y) \leq F^{-1}(t_{1-\alpha}|\mathbf{x}_{n+1}, D)\}.$$

EPICSCORE is summarized in Algorithm 1 and illustrated in Figure 3. Additional details on prior specification and strategies for managing its influence are provided in Appendix D.5. A discussion of the computational complexity introduced by incorporating Bayesian models into our framework is presented in Appendix D.6.

Algorithm 1: EPICSCORE

Input: Data $\mathcal{D} = \{(\mathbf{X}_i, Y_i)\}_{i=1}^n$, conformal score $s(\mathbf{X}, Y)$, nominal level α, test point $\mathbf{X}_{n+1}$

Step I: Fit conformal scores
1: Split data $\mathcal{D}$ into a training set $\mathcal{D}_{\text{train}}$ and a calibration set $\mathcal{D}_{\text{cal}}$.
2: Fit the conformal score $s(\mathbf{X}, Y)$ in $\mathcal{D}_{\text{train}}$.

Step II: Fit the predictive function
1: Split data $\mathcal{D}_{\text{cal}}$ into a training set $\mathcal{D}_{\text{cal},1}$ and a calibration set $\mathcal{D}_{\text{cal},2}$.
2. Fit predictive CDF $F(s|\mathbf{x}, D)$ using $\mathcal{D}_{\text{cal},1}$.
3. Compute EPICSCORE conformal score $s'(\mathbf{x}, y)$ for all elements of $\mathcal{D}_{\text{cal},2}$ (Eq. (2))
4: Compute the $(1 - \alpha)$ empirical quantile $t_{1-\alpha}$ of the conformal scores.

Step III: Compute prediction set
3: Compute the set $R_{\text{EPIC}}(\mathbf{X}_{n+1})$ as:

$$\begin{aligned} R_{\text{EPIC}}(\mathbf{X}_{n+1}) &= \{y \mid s'(\mathbf{X}_{n+1}, y) \leq t_{1-\alpha}\} \\ &= \{y \mid s(\mathbf{X}_{n+1}, y) \leq F^{-1}(t_{1-\alpha}|\mathbf{X}_{n+1}, D)\} \end{aligned}$$

2.2.1 Special Cases

We examine specific instances of conformal scores to provide further insight into how EPICSCORE captures epistemic uncertainty.

Regression. If the original conformal score is $s(\mathbf{x}, y) = |y - g(\mathbf{x})|$, the prediction regions given by EPICSCORE have the form

$$g(\mathbf{x}_{n+1}) \pm F^{-1}(t_{1-\alpha}|\mathbf{x}_{n+1}, D),$$

In particular, if $S|\mathbf{x}, D$ is modeled by a normal distribution with mean $\mu(\mathbf{x}, D)$ and standard deviation $\sigma(\mathbf{x}, D)$ the prediction sets will have the shape

$$(g(\mathbf{x}_{n+1}) - \mu(\mathbf{x}_{n+1}, D)) \pm \sigma(\mathbf{x}, D)\sqrt{2}\text{erf}^{-1}(2t_{1-\alpha} - 1),$$

where erf^{-1} denotes the inverse error function. This is equivalent to changing the original conformal score to $|y - g(\mathbf{x}_{n+1}) - \mu(\mathbf{x}_{n+1}, D)|/\sigma(\mathbf{x}_{n+1}, D)$, which is similar to the approach by Cocheteux et al. [2025], although

any process can be used to model the epistemic uncertainty in our version.

Quantile Regression. If the original conformal score s is given by Conformalized Quantile Regression (CQR) [Romano et al., 2019], the prediction regions of EPICSCORE have the form

$$\begin{aligned} &[q_{\alpha_1}(\mathbf{x}_{n+1}) - F^{-1}(t_{1-\alpha}|\mathbf{x}_{n+1}, D), \\ &q_{\alpha_2}(\mathbf{x}_{n+1}) + F^{-1}(t_{1-\alpha}|\mathbf{x}_{n+1}, D)]. \end{aligned}$$

Unlike the original CQR formulation, which expands or contracts the quantile regions $[q_{\alpha_1}(\mathbf{x}_{n+1}), q_{\alpha_2}(\mathbf{x}_{n+1})]$ by a constant factor t, EPICSCORE adjusts the regions dynamically based on the epistemic uncertainty at $\mathbf{x}_{n+1}$. This approach is similar, but more flexible than previous methods, such as UACQR-S [Rossellini et al., 2024], which imposes a correction factor of the form $t \times g(\mathbf{x}_{n+1})$. Also, any process can be used to model $F^{-1}(t_{1-\alpha}|\mathbf{x}_{n+1}, D)$.

Classification. Let $s(\mathbf{x}, y)$ be any nonconformity score for classification. One example is the APS score

$$s(\mathbf{x}, y) = \sum_{y' \in \mathcal{Y}: \widehat{\mathbb{P}}(y'|\mathbf{x}) > \widehat{\mathbb{P}}(y|\mathbf{x})} \widehat{\mathbb{P}}(y'|\mathbf{x}), \qquad (3)$$

where $\widehat{\mathbb{P}}(y'|\mathbf{x})$ represents the predicted probabilities from any classifier. Another common choice is

$$s(\mathbf{x}, y) = -\widehat{\mathbb{P}}(y|\mathbf{x}), \qquad (4)$$

[Vovk et al., 2005, Valle et al., 2023]. Since Y is discrete, the score $s(\mathbf{x}, Y)$ is also discrete. Moreover, the cumulative distribution function of its predictive distribution can be computed using the predictive distribution of the labels, $\mathbb{P}(y|\mathbf{x}, D)$. In particular, EPICSCORE is given by

$$\begin{aligned} s'(\mathbf{x}, y) &= \mathbb{P}\left(s(\mathbf{x}, Y) \leq s(\mathbf{x}, y)|\mathbf{x}, D\right) \\ &= \sum_{y'} \mathbb{I}(s(\mathbf{x}, y') \leq s(\mathbf{x}, y))\mathbb{P}(y'|\mathbf{x}, D) \\ &= \sum_{y': s(\mathbf{x}, y') \leq s(\mathbf{x}, y)} \mathbb{P}(y'|\mathbf{x}, D). \end{aligned}$$

This formulation reveals several key insights about EPICSCORE for classification:

- If the initial classifier $\widehat{\mathbb{P}}(y|\mathbf{x})$ is a neural network trained with Monte Carlo dropout or batch normalization, an approximation to $\mathbb{P}(y|\mathbf{x}, D)$ is readily available; one only needs to use the same technique at test time. This is because these methods provide a variational approximation of Bayesian predictive distributions [Gal and Ghahramani, 2016, Teye et al., 2018], eliminating the need to compute the predictive distribution using a separate holdout set.

- Both scoring functions in Eqs. 3 and 4 lead to the same EPICSCORE score. Additionally, EPICSCORE
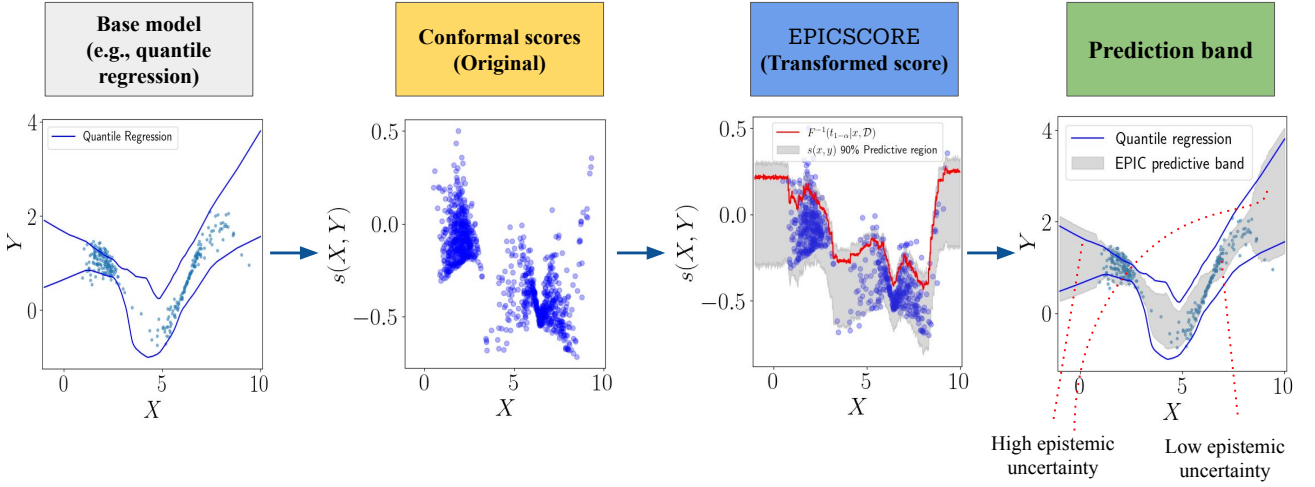
Figure 3: EPICSCORE schematic illustration: Given a fitted base model (first panel), we begin by creating a nonconformity score and evaluating it over the calibration set (second panel). We then model the predictive distribution of the conformal score $s(\mathbf{X}, Y)$ using a specified family of models, integrating the epistemic uncertainty about the data-generating process. The predictive CDF of each original score defines a new conformal score, allowing threshold computation in the transformed space (third panel). Finally, leveraging these predictive-based cutoffs, we construct an adaptive prediction band that accounts for epistemic uncertainty (fourth panel).

follows a similar structure to APS, with the key difference being that the estimates $\widehat{\mathbb{P}}(y|\mathbf{x})$ on the sum of Eq. 3 are replaced by the predictive distribution $\mathbb{P}(y|\mathbf{x}, D)$.

- For large D, no epistemic uncertainty remains and, therefore, EPICSCORE converges to the populational version of the APS score.

3 THEORY

Just like any split-conformal method, EPICSCORE guarantees marginal coverage as long as the data are exchangeable, regardless of the chosen Bayesian model (see Appendix E for all proofs):

Theorem 1. *Assuming that the data are exchangeable, the confidence region constructed by EPICSCORE satisfies marginal coverage, that is,*

$$\mathbb{P}\left(Y \in R_{EPIC}(\mathbf{X})\right) \geq 1 - \alpha.$$

Moreover, if the fitted scores follow a continuous joint distribution, the upper bound also holds:

$$\mathbb{P}\left(Y \in R_{EPIC}(\mathbf{X})\right) \leq 1 - \alpha + \frac{1}{1 + |\mathcal{D}_{cal,2}|}.$$

We now analyze its conditional coverage properties.

As the sample size of the calibration set used to compute the posterior D increases, the distribution function $F(s(\mathbf{x}, y)|\mathbf{x}, D)$ typically converges to $S(\mathbf{X}, Y)|\mathbf{x}, \theta^*$, where θ^* denotes the true parameter value [Schervish, 2012, Bernardo and Smith, 2009]. Consequently, EPICSCORE recovers the score proposed by Dheur et al. [2025, Eq. 14] in the limit of large calibration samples, which is exactly when epistemic uncertainty is negligible. This score is known to control asymptotic conditional coverage. We show that our proposed score exhibits the same property.

Formally, we assume that the predictive distribution converges to the true distribution of the conformal score [Bernardo and Smith, 2009, Schervish, 2012]:

Assumption 1. *For any $\varepsilon > 0$, we assume uniform convergence in probability over the randomness in D:*

$$\lim_{|D| \to \infty} \mathbb{P}\left(\sup_{s,\mathbf{x}} |F(s \mid \mathbf{x}, D) - F(s \mid \mathbf{x}, \theta^*)| > \varepsilon\right) = 0.$$

Next, we show that EPICSCORE has asymptotic conditional coverage:

Theorem 2. *Under Assumption 1, and assuming that the data are exchangeable, the confidence region constructed by EPICSCORE satisfies the asymptotic conditional coverage condition, that is:*

$$\lim_{|\mathcal{D}_{cal}| \to \infty} \mathbb{P}\left(Y \in R_{EPIC}(\mathbf{X}) \mid \mathbf{X} = \mathbf{x}\right) = 1 - \alpha.$$

Assumption 1 plays a key role in Theorem 2. While strong, it serves as a practical condition to ensure that the predictive distribution $F(s \mid \mathbf{x}, D)$ converges to the true distribution $F(s \mid \mathbf{x}, \theta^*)$ as the sample grows, so that the estimated quantiles approximate the ideal ones. Although potentially stronger than necessary, relaxing it would require a more technical analysis, which we leave for future work. Empirical results in Section 4 show that EPICSCORE achieves coverage close to the nominal level, suggesting robustness even when the assumption holds only approximately.

It is worth emphasizing that the type of guarantee provided by Theorem 2 is, in fact, the strongest achievable in this setting. As shown by Lei and Wasserman [2014], exact conditional coverage is unattainable in general conformal frameworks. Recent methods aim to improve local coverage [Boström and Johansson, 2020, Foygel Barber et al., 2021, Cabezas et al., 2025] and could be combined with EPICSCORE to improve local calibration.

4 EXPERIMENTS

In this section, we evaluate our framework against state-of-the-art baselines, applying EPICSCORE to two initial conformal scores: (i) quantile-based and (ii) regression-based. Each version of EPICSCORE is compared to appropriate baselines, which are detailed in the following subsections. In Appendix D.7 we also show that EPICSCORE can be applied to density-based scores.

We consider three versions of EPICSCORE, each using a different model for the predictive distribution (see implementation details in Appendix D):

- **Bayesian Additive Regression Trees** [Chipman et al., 2012]: The BART model represents the score as a sum of regression trees:

$$s(Y, \mathbf{X})|\mathbf{X}, \boldsymbol{\theta} \sim \phi\left(\sum_{i=1}^{m} G_i(\mathbf{X}, T_i, M_i), \sigma\right),$$

where ϕ denotes a probability distribution, σ its associated scale, G_i a binary tree with structure T_i and leaf values M_i. We set ϕ as a Normal distribution and σ depending on $\mathbf{X}$ to incorporate heteroskedasticity [Pratola et al., 2020].

- **Gaussian Process (GP)** [Williams and Rasmussen, 2006, Schulz et al., 2018]: For the GP regression model, we assume the score follows the form $s(Y, \mathbf{x}) = f(\mathbf{x}) + \varepsilon$, with $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ representing independent Gaussian noise, and $f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ is a Gaussian Process with mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$. We adopt variational approximations to the predictive distribution [Salimbeni et al., 2018], which offer scalability.

- **Mixture Density Network with MC-Dropout [Bishop, 1994, Gal and Ghahramani, 2016]**: The Mixture Density Network (MDN) models the score distribution using a weighted sum of Gaussian components:

$$f(s(y, \mathbf{x})|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) N(s(y, \mathbf{x})|\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})),$$

where $N(\cdot)$ denotes the normal density and $\pi_k(\cdot)$, $\mu_k(\cdot)$, $\sigma_k(\cdot)$ are all modeled by neural networks, with $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$. To derive a predictive distribution for the scores, we incorporate dropout at each MDN layer. By performing multiple stochastic forward passes using MC Dropout, we approximate the posterior distribution of the MDN parameters, thus propagating uncertainty into the predictive score distribution [Gal and Ghahramani, 2016]. Appendix B explains how MC Dropout yields predictive distributions.

Our comparisons are conducted using 13 datasets commonly employed for benchmarking in the conformal prediction literature: Airfoil [Dua et al., 2017], Bike [Mehra, 2023], Concrete [Yeh, 1998, Dua et al., 2017], Cycle [Tfekci and Kaya, 2014, Dua et al., 2017], HomesKaggle [2016], Electric [Dua et al., 2017], Meps19 [Romano et al., 2019], Protein [Rana, 2013, Dua et al., 2017], Star [Achilles et al., 2008], Super-Conductivity [Hamidieh, 2018], WEC [Neshat et al., 2020], WineRed [Cortez et al., 2009], and WineWhite [Cortez et al., 2009]. Additional details on these datasets are provided in Table 5 of the Appendix.

We report the average performance across 50 runs, highlighting methods that achieve statistically significant improvements based on 95% confidence interval of each evaluation metric. In each run, we randomly partition the data into 40% for training, 40% for calibration, and 20% for testing.

We use the Average Interval Score Loss (AISL) [Gneiting and Raftery, 2007] as our main evaluation metric, as it balances coverage and interval length, promoting narrower yet valid prediction intervals. To evaluate how each method captures epistemic uncertainty, we also report the outlier-to-inlier interval length ratio and average coverage on outliers, which measure adaptivity in data-scarce regions. Additionally, we include (i) average interval length, (ii) marginal coverage, and (iii) the Pearson correlation between coverage and interval length, which serves as a proxy for conditional coverage quality [Feldman et al., 2021]. A full description of all metrics can be found in Appendix C.1, with extended empirical results presented in Appendix C.2.

4.1 QUANTILE-REGRESSION BASELINES

For quantile regression-based scores, we adopt CatBoost [Dorogush et al., 2018] as the base quantile-regression

Table 1: Quantile regression AISL values for each method and dataset. The table reports the mean across 50 runs, with twice the standard deviation in brackets. Bold values indicate the best-performing method within a 95% confidence interval. `EPICSCORE` demonstrates strong performance across most datasets and consistently ranks among the top methods.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	CQR	CQR-r	UACQR-P	UACQR-S
airfoil	19.361 (0.234)	19.704 (0.27)	**18.799 (0.29)**	20.521 (0.234)	20.535 (0.236)	23.021 (0.337)	20.188 (0.3)
bike $\times(10^1)$	44.722 (0.297)	47.818 (0.320)	**43.858 (0.326)**	45.628 (0.256)	45.638 (0.258)	53.413 (0.376)	**43.815 (0.385)**
concrete	**42.765 (0.723)**	45.276 (0.764)	44.442 (0.8)	46.882 (0.681)	46.896 (0.683)	52.789 (1.097)	47.324 (1.349)
cycle	34.435 (0.142)	35.054 (0.131)	**34.077 (0.129)**	39.218 (0.134)	39.408 (0.136)	43.775 (0.181)	35.346 (0.197)
electric	0.099 (<0.001)	0.096 (<0.001)	**0.082 (<0.001)**	0.102 (0.001)	0.102 (0.001)	0.111 (0.001)	0.097 (<0.001)
homes $\times(10^5)$	7.739 (0.066)	8.098 (0.072)	**7.225 (0.049)**	8.360 (0.075)	8.433 (0.078)	11.427 (0.131)	8.544 (0.107)
meps19	**65.085 (1.469)**	**64.907 (1.56)**	**64.3 (1.528)**	**64.239 (1.56)**	**64.239 (1.56)**	71.015 (1.763)	**63.737 (1.461)**
protein	17.687 (0.019)	18.096 (0.037)	**17.417 (0.019)**	17.7 (0.015)	17.7 (0.016)	18.149 (0.015)	17.691 (0.015)
star $\times(10^1)$	**98.466 (0.768)**	98.033 (0.750)	98.725 (0.754)	97.770 (0.725)	97.791 (0.724)	99.782 (0.647)	99.809 (0.968)
superconductivity	74.37 (0.222)	80.278 (0.266)	**70.212 (0.196)**	75.496 (0.219)	75.508 (0.218)	87.929 (0.513)	73.971 (0.404)
WEC $\times(10^5)$	2.925 (0.009)	2.665 (0.012)	**2.374 (0.010)**	3.138 (0.009)	3.142 (0.009)	3.517 (0.010)	3.046 (0.010)
winered	**3.007 (0.058)**	**3.009 (0.059)**	**2.977 (0.05)**	**2.979 (0.069)**	**2.978 (0.069)**	3.059 (0.069)	**2.999 (0.063)**
winewhite	3.334 (0.03)	3.327 (0.034)	**3.219 (0.03)**	3.316 (0.036)	3.315 (0.036)	3.378 (0.038)	**3.2 (0.036)**

Table 2: Regression AISL values for each method and dataset. The reported values represent the average across 50 runs, with two times the standard deviation in parentheses. Bolded values highlight the method with superior performance within a 95% confidence interval. `EPICSCORE` demonstrates competitive or superior performance compared to other methods.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	Mondrian	Reg-split	Weighted
airfoil	**19.747 (0.767)**	**20.287 (0.686)**	**19.823 (0.675)**	21.532 (0.919)	21.201 (0.98)	**20.276 (0.819)**
bike $\times(10^1)$	**36.381 (0.463)**	41.448 (0.575)	**37.041 (0.452)**	38.190 (0.403)	43.918 (0.567)	37.773 (0.468)
concrete	**52.098 (2.237)**	**52.998 (2.359)**	**51.648 (2.185)**	61.915 (2.815)	**54.902 (2.634)**	58.399 (3.165)
cycle	**19.418 (0.211)**	19.522 (0.221)	**19.436 (0.213)**	**19.403 (0.226)**	19.73 (0.208)	**19.49 (0.207)**
electric	**0.048 (<0.001)**	0.049 (<0.001)	**0.048 (<0.001)**	0.05 (<0.001)	0.05 (0.001)	**0.048 (<0.001)**
homes $\times(10^5)$	5.921 (0.0716)	6.192 (0.0689)	**5.546 (0.055)**	5.710 (0.053)	7.569 (0.098)	5.860 (0.056)
meps19	86.039 (2.421)	87.086 (2.405)	**75.061 (1.807)**	79.192 (1.821)	109.83 (2.695)	92.433 (3.259)
protein	18.885 (0.054)	18.772 (0.065)	17.735 (0.055)	**17.586 (0.051)**	19.423 (0.055)	18.314 (0.065)
star $\times(10^1)$	**105.616 (1.255)**	**106.112 (0.998)**	106.368 (1.173)	109.346 (1.119)	**105.250 (1.038)**	129.492 (1.657)
superconductivity	54.895 (0.364)	59.16 (0.449)	**53.406 (0.365)**	58.065 (0.313)	68.183 (0.418)	54.981 (0.345)
WEC $\times(10^5)$	1.437 (0.010)	1.435 (0.011)	**1.283 (0.009)**	**1.294 (0.009)**	1.620 (0.009)	1.410 (0.009)
winered	**3.152 (0.07)**	**3.171 (0.064)**	**3.101 (0.062)**	3.262 (0.069)	**3.214 (0.063)**	3.415 (0.067)
winewhite	**3.104 (0.027)**	3.187 (0.029)	**3.129 (0.029)**	**3.087 (0.023)**	3.181 (0.028)	3.189 (0.033)

model in all conformal methods. See Appendix C.3 for details on hyperparameters. We compare `EPICSCORE` to the following baselines:

- **CQR** [Romano et al., 2019], the conformal quantile regression method described in the introduction.

- **CQR-r** [Sesia and Candès, 2020], which scales each derived cutoff by the interval width to produce adaptive intervals. As CQR, this approach accounts only for aleatoric uncertainty.

- **UACQR-P** and **UACQR-S** [Rossellini et al., 2024], that integrate epistemic uncertainty through ensemble-based statistics. We use their default strategies: UACQR-S with ensemble standard deviation, and UACQR-P with ensemble order statistics.

All baselines are fitted and evaluated using the implementation from Rossellini et al. [2024].

4.2 REGRESSION BASELINES

For regression-based conformal scores, we use a neural network optimized with a penalized Mean Squared Loss. Detailed descriptions of the architectures and hyperparameters used can be found in Appendix C.3. We compare `EPICSCORE` to the following baselines:

- **Regression Split** [Lei and Wasserman, 2014], the conformal method based on residuals from a regression model described in the introduction.

- **Weighted Regression Split** [Lei et al., 2018], which multiplies the derived cutoff by a conditional Mean Absolute Deviance (MAD) estimate to yield adaptive intervals. The MAD is modeled by regressing the training set's absolute residuals on $\mathbf{X}$, using the same model architecture as the base predictor.

- **Mondrian Conformal Regression** [Boström and Johansson, 2020], which enhances conditional coverage by adaptively partitioning the feature space using a binning scheme based on conditional variance. We es-

Table 3: Average outlier-to-inlier interval length ratio across methods and datasets in the quantile regression setting. This table reports the average ratio between prediction interval lengths for outliers and inliers across 50 runs. Bold entries denote the best-performing method within a 95% confidence interval. Overall, `EPICSCORE` consistently achieves higher ratios, demonstrating its effectiveness in adjusting interval widths based on data sparsity.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	CQR	CQR-r	UACQR-P	UACQR-S
airfoil	**1.028 (0.034)**	**1.01 (0.018)**	**1.048 (0.04)**	**1.005 (0.013)**	**1.004 (0.012)**	**1.002 (0.01)**	**1.005 (0.016)**
bike	0.991 (0.007)	**1.01 (0.007)**	0.989 (0.007)	0.985 (0.007)	0.985 (0.007)	0.99 (0.005)	0.981 (0.01)
concrete	0.983 (0.033)	0.995 (0.016)	**1.043 (0.029)**	0.998 (0.012)	0.998 (0.012)	0.998 (0.008)	0.998 (0.012)
cycle	0.953 (0.014)	0.966 (0.014)	0.949 (0.014)	**0.997 (0.005)**	**0.997 (0.005)**	**1.0 (0.004)**	**0.998 (0.007)**
electric	**1.01 (0.004)**	**1.01 (0.004)**	**1.012 (0.008)**	1.008 (0.004)	1.007 (0.003)	1.006 (0.003)	**1.01 (0.005)**
homes	1.116 (0.019)	1.101 (0.016)	**1.173 (0.027)**	1.088 (0.015)	1.081 (0.014)	1.039 (0.006)	1.094 (0.014)
meps19	**1.052 (0.029)**	**1.042 (0.025)**	**1.044 (0.028)**	**1.04 (0.025)**	**1.04 (0.025)**	1.021 (0.015)	**1.045 (0.026)**
protein	1.001 (0.002)	1.001 (0.003)	**1.006 (0.002)**	1.002 (0.001)	1.002 (0.001)	1.001 (0.001)	1.002 (0.001)
star	**0.994 (0.006)**	**0.994 (0.003)**	**0.999 (0.007)**	**0.994 (0.003)**	0.993 (0.003)	**0.997 (0.003)**	**0.997 (0.003)**
superconductivity	1.021 (0.007)	0.987 (0.006)	1.029 (0.008)	**1.041 (0.007)**	**1.041 (0.007)**	1.023 (0.004)	**1.049 (0.008)**
WEC	**1.012 (0.006)**	1.003 (0.008)	**1.02 (0.01)**	1.016 (0.003)	1.016 (0.003)	1.011 (0.002)	**1.021 (0.004)**
winered	1.008 (0.012)	1.007 (0.008)	**1.072 (0.022)**	0.993 (0.008)	0.992 (0.008)	0.996 (0.005)	0.992 (0.009)
winewhite	1.031 (0.007)	1.024 (0.006)	**1.053 (0.014)**	1.018 (0.007)	1.018 (0.007)	1.018 (0.006)	**1.034 (0.01)**

timate variance by fitting a Random Forest to $(\mathbf{X}, Y)$.

4.3 RESULTS

The mean average coverage is close to the nominal 90% for all methods (Table 6 and 10 of Appendix C.2), which is expected since all methods are conformal.

In the quantile regression setting, Table 1 highlights `EPICSCORE`'s strong performance across all datasets, with the MDN-MC Dropout variant ranking among the top in 12 out of 13 cases. Notably, `EPICSCORE` outperforms all competing methods on 7 datasets, effectively balancing coverage and interval sharpness. Table 3 demonstrates `EPICSCORE`'s superior adaptability to data-sparse regions, with the MDN-MC Dropout variant achieving the highest outlier-to-inlier interval length ratio in 10 out of 13 cases. Table 9 further confirms good coverage for outliers in those same sparse regions. Additionally, Tables 7 and 8 (Appendix C.2) show that this variant yields narrower intervals while maintaining strong conditional coverage.

For the regression setting, Table 2 confirms `EPICSCORE`'s strong performance across most datasets, with the MDN-MC Dropout variant excelling in 12 out of 13 cases and the BART version ranking among the top in 8. Tables 13 and 14 report the outlier-to-inlier interval length ratio and outlier coverage, respectively—both showing that the MDN-MC Dropout variant performs consistently well in data-sparse regions. Additionally, Tables 11 and 12 (Appendix C.2) confirm its ability to generate concise intervals while preserving approximate conditional coverage.

5 FINAL REMARKS

We introduce `EPICSCORE`, a novel conformal score that incorporates epistemic uncertainty into predictive regions. Us-

ing Bayesian modeling, `EPICSCORE` dynamically adjusts any nonconformity score to account for epistemic uncertainty, ensuring coverage even in sparse regions. We prove it preserves marginal coverage and achieves asymptotic conditional coverage. Empirical results show `EPICSCORE` often outperforms alternatives, producing prediction regions that better reflect uncertainty while maintaining valid coverage.

Unlike previous approaches that rely on specific modeling choices or task-dependent formulations, `EPICSCORE` is fully model-agnostic. Any Bayesian model can estimate the epistemic uncertainty of a given conformal score, allowing practitioners to tailor the method to their application. This flexibility extends `EPICSCORE`'s applicability across regression, classification, and structured prediction problems.

Looking ahead, we plan to extend `EPICSCORE` to settings involving distribution shift, where accounting for epistemic uncertainty is important due to increased data sparsity in certain regions. In addition, we intend to improve the efficiency of our calibration procedure by eliminating the need to split the calibration set. To this end, we will explore adaptations based on Jackknife+ and CV+ methods [Barber et al., 2021], as well as multiple-split conformal prediction techniques [Lei et al., 2018], which offer ways to aggregate information with statistical guarantees. Code to implement `EPICSCORE` and reproduce the experiments is available at `https://github.com/Monoxido45/EPICSCORE`.

References

Charles M Achilles, Helen Pate Bain, Fred Bellott, Jayne Boyd-Zaharias, Jeremy Finn, John Folger, John Johnston, and Elizabeth Word. Tennessee's student teacher achievement ratio (star) project. *Harvard Dataverse*, 1:2008, 2008.

Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. Wiley Series in Probability and Statistics. Wiley, 2009. ISBN 9780470317716. URL https://books.google.com.br/books?id=11nSgIcd7xQC.

Christopher M Bishop. Mixture density networks. 1994.

Henrik Boström and Ulf Johansson. Mondrian conformal regressors. In *Conformal and Probabilistic Prediction and Applications*, pages 114–133. PMLR, 2020.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

Luben Cabezas, Guilherme P Soares, Thiago R Ramos, Rafael B Stern, and Rafael Izbicki. Distribution-free calibration of statistical confidence sets. *arXiv preprint arXiv:2411.19368*, 2024.

Luben MC Cabezas, Mateus P Otto, Rafael Izbicki, and Rafael B Stern. Regression trees for fast and adaptive prediction intervals. *Information Sciences*, 686:121369, 2025.

Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021.

Hugh A Chipman, Edward I George, and Robert E McCulloch. BART: Bayesian additive regression trees. *Annals of Applied Statistics*, 6(1):266–298, 2012.

Mathieu Cocheteux, Julien Moreau, and Franck Davoine. Uncertainty-aware online extrinsic calibration: A conformal prediction approach. *arXiv preprint arXiv:2501.06878*, 2025.

Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

Daniel Csillag, Lucas Monteiro Paes, Thiago Ramos, João Vitor Romano, Rodrigo Schuller, Roberto B Seixas, Roberto I Oliveira, and Paulo Orenstein. Amnioml: amniotic fluid segmentation and volume prediction with uncertainty quantification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15494–15502, 2023.

Niccolò Dalmasso, Luca Masserano, David Zhao, Rafael Izbicki, and Ann B Lee. Likelihood-free frequentist inference: Bridging classical statistics and machine learning for reliable simulator-based inference. *Electronic Journal of Statistics*, 18(2):5045–5090, 2024.

Victor Dheur, Matteo Fontana, Yorick Estievenart, Naomi Desobry, and Souhaib Ben Taieb. Multi-output conformal regression: A unified comparative study with new conformity scores. *arXiv preprint arXiv:2501.10533*, 2025.

Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

Dheeru Dua, Casey Graff, et al. Uci machine learning repository, 2017. *URL http://archive. ics. uci. edu/ml*, 7 (1):62, 2017.

Shai Feldman, Stephen Bates, and Yaniv Romano. Improving conditional coverage via orthogonal quantile regression. *Advances in neural information processing systems*, 34:2060–2071, 2021.

Edwin Fong and Chris C Holmes. Conformal bayesian computation. *Advances in Neural Information Processing Systems*, 34:18268–18279, 2021.

Matteo Fontana, Gianluca Zeni, and Simone Vantini. Conformal prediction: a unified review of theory and new challenges. *Bernoulli*, 29(1):1–23, 2023.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

Peter I Frazier. Bayesian optimization. In *Recent advances in optimization and modeling of contemporary problems*, pages 255–278. Informs, 2018.

Peter E Freeman, Rafael Izbicki, and Ann B Lee. A unified framework for constructing, tuning and assessing photometric redshift density estimates in a selection bias setting. *Monthly Notices of the Royal Astronomical Society*, 468 (4):4556–4565, 2017.

Alek Fröhlich, Thiago Ramos, Gustavo Motta Cabello Dos Santos, Isabela Panzeri Carlotti Buzatto, Rafael Izbicki, and Daniel Guimaraes Tiezzi. Personalizedus: Interpretable breast cancer risk assessment with local coverage uncertainty quantification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27998–28006, 2025.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1050–1059. PMLR, 2016.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Irving John Good. The bayes/non-bayes compromise: A brief review. *Journal of the American Statistical Association*, 87(419):597–606, 1992.

Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50, 2023.

Kam Hamidieh. Superconductivty Data. UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C53P47.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning. *arXiv preprint arXiv:2302.13425*, 2023.

Peter Hoff. Bayes-optimal prediction with frequentist coverage control. *Bernoulli*, 29(2):901–928, 2023.

Ana Horta, B Malone, U Stockmann, B Minasny, TFA Bishop, AB McBratney, R Pallasser, and L Pozza. Potential of integrated field spectroscopy and spatial analysis for enhanced assessment of soil contamination: A prospective review. *Geoderma*, 241:180–209, 2015.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110: 457–506, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

Rafael Izbicki. *Machine Learning Beyond Point Predictions: Uncertainty Quantification*. 1st edition, 2025. ISBN 978-65-01-20272-3.

Rafael Izbicki and Ann B Lee. Converting high-dimensional regression to high-dimensional conditional density estimation. *Electronic Journal of Statistics*, 11:2800–2831, 2017.

Rafael Izbicki, Gilson Shimizu, and Rafael Stern. Flexible distribution-free conditional predictive bands using density estimators. In *International Conference on Artificial Intelligence and Statistics*, pages 3068–3077. PMLR, 2020.

Rafael Izbicki, Gilson Shimizu, and Rafael B Stern. Cd-split and hpd-split: Efficient conformal regions in high dimensions. *Journal of Machine Learning Research*, 23 (87):1–32, 2022.

Edgar Jaber, Vincent Blot, Nicolas Brunel, Vincent Chabridon, Emmanuel Remy, Bertrand Iooss, Didier Lucor, Mathilde Mougeot, and Alessandro Leite. Conformal approach to gaussian process surrogate evaluation with coverage guarantees. *arXiv preprint arXiv:2401.07733*, 2024.

Kaggle. House sales in king county, usa. Online, 2016. URL https://www.kaggle.com/harlfoxem/housesalesprediction/metadata. Accessed: August, 2019.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Siddharth Krishna Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017.

Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76 (1):71–96, 2014.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Valery Manokhin. Awesome conformal prediction, 2024. URL https://github.com/valeman/awesome-conformal-prediction.

Luca Masserano, Tommaso Dorigo, Rafael Izbicki, Mikael Kuusela, and Ann Lee. Simulator-based inference with waldo: Confidence regions by leveraging prediction algorithms and posterior estimators for inverse problems. In *International Conference on Artificial Intelligence and Statistics*, pages 2960–2974. PMLR, 2023.

Rowan Thomas McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017.

Raj Mehra. Bike sharing demand - rmsle: 0.3194. https://www.kaggle.com/code/rajmehra03/bike-sharing-demand-rmsle-0-3194/input?select=train.csv, 2023. Accessed: [Insert Date Here].

Zhibao Mian, Xiaofei Deng, Xiaohui Dong, Yuzhu Tian, Tianya Cao, Kairan Chen, and Tareq Al Jaber. A literature review of fault diagnosis based on ensemble learning. *Engineering Applications of Artificial Intelligence*, 127: 107357, 2024.

Mehdi Neshat, Bradley Alexander, Nataliia Y Sergiienko, and Markus Wagner. Optimisation of large wave farms using a multi-strategy evolutionary framework. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 1150–1158, 2020.

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

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Aurélien Pion and Emmanuel Vazquez. Gaussian process interpolation with conformal prediction: methods and comparative analysis. In *LOD 2024, 10th International Conference on Machine Learning, Optimization, and Data Science*, 2024.

Vincent Plassier, Alexander Fishkov, Maxim Panov, and Eric Moulines. Conditionally valid probabilistic conformal prediction. *stat*, 1050:1, 2024.

Matthew T Pratola, Hugh A Chipman, Edward I George, and Robert E McCulloch. Heteroscedastic bart via multiplicative regression trees. *Journal of Computational and Graphical Statistics*, 29(2):405–417, 2020.

Miriana Quiroga, Pablo G Garay, Juan M Alonso, Juan Martin Loyola, and Osvaldo A Martin. Bayesian additive regression trees for probabilistic programming. *arXiv preprint arXiv:2206.03619*, 2022.

Prashant Rana. Physicochemical Properties of Protein Tertiary Structure. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5QW3H.

Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, pages 3543–3553. Curran Associates, Inc., 2019.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

Raphael Rossellini, Rina Foygel Barber, and Rebecca Willett. Integrating uncertainty awareness into conformalized quantile regression. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1548. PMLR, 2024.

Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In *International Conference on Artificial Intelligence and Statistics*, pages 689–697. PMLR, 2018.

Mark J Schervish. *Theory of statistics*. Springer Science & Business Media, 2012.

S. J. Schmidt, A. I. Malz, J. Y. H. Soo, I. A. Almosallam, M. Brescia, S. Cavuoti, J. Cohen-Tanugi, A. J. Connolly, J. DeRose, P. E. Freeman, M. L. Graham, K. G. Iyer, M. J. Jarvis, J. B. Kalmbach, E. Kovacs, A. B. Lee, G. Longo, C. B. Morrison, J. A. Newman, E. Nourbakhsh, E. Nuss, T. Pospisil, H. Tranin, R. H. Wechsler, R. Zhou, R. Izbicki, and LSST Dark Energy Science Collaboration. Evaluation of probabilistic photometric redshift estimation approaches for The Rubin Observatory Legacy Survey of Space and Time (LSST). *Monthly Notices of the Royal Astronomical Society*, 499(2):1587–1606, December 2020.

Eric Schulz, Maarten Speekenbrink, and Andreas Krause. A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of mathematical psychology*, 85:1–16, 2018.

Raghavendra Selvan, Frederik Faye, Jon Middleton, and Akshay Pai. Uncertainty quantification in medical image segmentation with normalizing flows. In *Machine Learning in Medical Imaging: 11th International Workshop, MLMI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings 11*, pages 80–90. Springer, 2020.

Matteo Sesia and Emmanuel J Candès. A comparison of some conformal quantile regression methods. *Stat*, 9(1): e261, 2020.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *International Conference on Machine Learning*, pages 4907–4916. PMLR, 2018.

Pnar Tfekci and Heysem Kaya. Combined Cycle Power Plant. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5002N.

Hristos Tyralis and Georgia Papacharalampous. A review of predictive uncertainty estimation with machine learning. *Artificial Intelligence Review*, 57(4):94, 2024.

D. Valle, R. Leite, Rafael Izbicki, C. Silva, and L. Haneda. Local uncertainty maps for land-use/land-cover classification without remote sensing and modeling work using a class-conditional conformal approach. *International Journal of Applied Earth Observation and Geoinformation*, 2024.

Denis Valle, Rafael Izbicki, and Rodrigo Vieira Leite. Quantifying uncertainty in land-use land-cover classification using conformal statistics. *Remote Sensing of Environment*, 295:113682, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008.

Janette Vazquez and Julio C Facelli. Conformal prediction in clinical medical sciences. *Journal of Healthcare Informatics Research*, 6(3):241–252, 2022.

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

Tianyang Wang, Yunze Wang, Jun Zhou, Benji Peng, Xinyuan Song, Charles Zhang, Xintian Sun, Qian Niu, Junyu Liu, Silin Chen, et al. From aleatoric to epistemic: Exploring uncertainty quantification techniques in artificial intelligence. *arXiv preprint arXiv:2501.03282*, 2025.

Larry Wasserman. Frasian inference. 2011.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5PK67.

Epistemic Uncertainty in Conformal Scores: A Unified Approach (Appendix)

Luben M. C. Cabezas[1, 2]   **Vagner S. Santos**[1]   **Thiago R. Ramos**[1]   **Rafael Izbicki**[1]

[1]Department of Statistics, Federal University of São Carlos, São Carlos, São Paulo, Brazil
[2]Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, São Paulo, Brazil

A   TECHNICAL DETAILS AND SUPPLEMENTARY RESULTS FOR THE INTRODUCTION'S EXAMPLES

A.1   REGRESSION

In this section, we detail the example presented by Figure 1. We simulate a scenario with two distinct dense regions exhibiting low aleatoric and epistemic uncertainty, separated by an intermediate, sparser region with high aleatoric and epistemic uncertainty. Given a sample size n, we first generate $\lfloor 0.425 \cdot n \rfloor$ samples for each $X \sim U(0, 1.5)$ and $X \sim U(8, 10)$, ensuring that 85% of the data comes from the two outer regions, reflecting low epistemic uncertainty. The corresponding response variable follows $Y \sim N(2 \sin X, 0.1)$, which also reflects low aleatoric uncertainty. For the remaining $\lfloor 0.15 \cdot n \rfloor$ samples, we draw X from a transformed Beta distribution, $X \sim (\text{Beta}(8, 8) \cdot (8 - 1.5)) + 1.5$, concentrating points in the intermediate region. Here, the response variable follows $Y \sim N(2 \sin X, 2.1)$, introducing high aleatoric uncertainty. Epistemic uncertainty is particularly elevated at the boundaries of this region.

In this setting, we use a K-nearest neighbors (KNN) algorithm with $k = 10$ as the regression base model $g(\mathbf{x})$. We illustrate the difference between EPICSCORE and established conformal prediction baselines for regression intervals, including Regression Split [Lei and Wasserman, 2014], Weighted Regression Split [Lei et al., 2018], and the Mondrian Conformal Regression [Boström and Johansson, 2020]. For EPICSCORE, we use the BART-based version, with $m = 100$ trees, default prior options for all parameters, and a heteroscedastic gamma distribution as the probability model for the conformal score (detailed in D.4), which is appropriate given that the regression conformal score is non-negative and often asymmetric.

In terms of baselines, both the Weighted and Mondrian methods estimate the conditional spread to construct prediction intervals. The locally weighted approach models the Mean Absolute Deviance (MAD), $\mathbb{E}[|g(\mathbf{X}) - Y||\mathbf{X}]$, by regressing absolute residuals $|g(\mathbf{x}) - y|$ on $\mathbf{X}$ using the same model type as $g(\mathbf{x})$. Meanwhile, the Mondrian method partitions the feature space using a binning scheme (or taxonomy) based on an estimation of conditional variance $\mathbb{V}[Y|\mathbf{X}]$, generally obtained using ensemble-based variance, commonly derived from an additionally fitted Random Forest.

Visually, both baseline methods outperform regression split, providing adaptive prediction intervals that widen in regions with high aleatoric uncertainty and narrow in regions with low aleatoric uncertainty. However, both methods struggle to generate wider intervals in data-sparse regions, such as $x \in (1.5, 2)$ and $x \in (7, 8)$. This limitation arises because these regions have low spread estimates due to insufficient data, leading to underestimated cutoffs. In contrast, EPICSCORE offers widened predictive intervals in these regions, better capturing epistemic uncertainty, while still accurately representing uncertainty in data-rich areas.

A.2   CLASSIFICATION

For the image classification example (introduced in Figure 2), we used the publicly available CIFAR-100 dataset [Krizhevsky et al., 2009], which consists of $60,000$ color images of size 32×32 spanning 100 classes, with each class containing 600 images. The dataset was split into training, calibration, and test sets, allocating 10% for testing and 45% each for training

and calibration. We utilized a ResNet-34 model [He et al., 2016] to extract 512-dimensional feature representations and trained a Random Forest classifier on the training set with default parameters except for the number of trees, which we fixed at 300. The classifier reports an accuracy of 47% in the test set. The Adaptive Prediction Set and `EPICSCORE` were then applied to the calibration set, using the MDN MC-dropout variant of `EPICSCORE`, which model's architecture is better detailed in D.2.

In the classification setting, `EPICSCORE` includes an alternative adaptation to handle the discrete nature of the conformal score (see Section 2.2.1 for details). However, it can also be applied similarly to the regression and quantile regression settings by treating the conformal score as continuous, fitting a predictive distribution, and deriving adaptive cutoffs. This approach involves normalizing the score to better leverage the chosen models. Given the large number of classes in CIFAR-100, this approximation remains valid, as it produces more fine-grained probability vectors. In this example, we adopt this formulation of `EPICSCORE`, using the APS score as the conformal score and deriving adaptive thresholds from its fitted predictive distribution.

To provide both a broad and detailed performance comparison, we first assess each method's coverage and set size using the size-stratified coverage (SSC) metric [Angelopoulos and Bates, 2021]. Next, we examine and visualize the prediction set sizes for the top 150 outliers and inliers, evaluating how well each method captures epistemic uncertainty in outliers while remaining adaptive for inliers. For illustration purposes, we set $\alpha = 0.2$ across all analyses.

SSC metric

The SSC metric aims to evaluate the calibration of prediction sets by stratifying them into G bins $\{B_j\}_{j=1}^{G}$ based on their cardinality. For $j < G$, the bin B_j contains the prediction sets with cardinality j, while B_G includes all sets with at least G elements. Formally, let $I_j = \{i : R(\mathbf{X}_i) \in B_j\}$ denote the indices of prediction sets that fall into bin B_j. The SSC metric for a given prediction set method $R(\cdot)$ is then defined as:

$$SSC(R) = \min_{j \in \{1, \ldots, G\}} \frac{1}{|I_j|} \sum_{i \in I_j} \mathbb{I}\{Y_i \in R(\mathbf{X}_i)\} \ . \tag{5}$$

Intuitively, this metric measures the minimum coverage of $R(\cdot)$ across different set sizes, assessing whether coverage remains stable despite changes in set cardinality. SSC values close to $1 - \alpha$ indicate strong coverage performance, while values farther from $1 - \alpha$ suggest greater violations of conditional coverage [Angelopoulos and Bates, 2021]. For this analysis, we set $G = 15$. Table 4 reports the SSC average values and 2 times the standard error across 10 runs for both methods.

Table 4: Average SSC metric over 10 runs, with twice the standard deviation in brackets. `EPICSCORE` achieves SSC values closer to the nominal level compared to APS.

Method	SSC	2 * SE
`EPICSCORE`-MDN	0.734	0.014
APS	0.553	0.036

These results show that `EPICSCORE`-MDN achieves an SSC much closer to the nominal level of 0.8 than APS, indicating more consistent coverage across different set sizes. In contrast, APS has a lower average SSC, reflecting greater deviations from target coverage and potentially less reliable uncertainty quantification.

Outlier and inlier analysis

To differentiate inliers from outliers for further analysis and comparison of each method's prediction sets, we first apply t-SNE [Van der Maaten and Hinton, 2008] to reduce the dimensionality of the feature space in the test set. We then use the Local Outlier Factor (LOF) method [Breunig et al., 2000] for outlier detection, leveraging a KNN-based density estimation to identify anomalies. The LOF score not only highlights outliers but also helps characterize typical (inlier) observations, offering a structured approach to assessing epistemic uncertainty across different regions of the data distribution. We fit LOF on the first two t-SNE dimensions, assuming a contamination rate of 10% (i.e., treating 10% of the sample as outliers), and rank the top 150 outliers and inliers for analysis.

In general, we expect outliers to have wider prediction sets, as they are located in sparser regions of the feature space. In contrast, inliers are likely to have narrower sets, reflecting their position in denser, more typical regions of the data distribution. Beyond Figure 2, Figures 4 and 5 provide additional examples that further illustrate this behavior, emphasizing how EPICSCORE differentiates itself from APS. Additionally, Figure 6 displays the distribution of set sizes for outliers and inliers across both methods. While both methods generate larger prediction sets for outliers than for inliers, EPICSCORE shows higher set sizes and a more dispersed distribution for outliers compared to APS, while presenting a more concentrated distribution for inliers.

Furthermore, we observe that the APS set sizes do not exceed a cardinality of 20, highlighting its lack of adaptability. Overall, these results reinforce the flexibility of our method across regions with varying levels of epistemic uncertainty, showcasing its advantage over the standard APS by explicitly incorporating epistemic uncertainty into the cutoff derivation.

True label: cockroach

APS set: baby, beaver, bee, **cockroach**,
crab, hamster, lobster, table

EPIC set: baby, beaver, bee, beetle, boy,
butterfly, **cockroach**, crab, hamster,
lawn_mower, lobster, plate, rabbit, rose, table,
woman

True label: sweet_pepper

APS set: apple, bowl, can, clock, pear, plate,
sweet_pepper

EPIC set: apple, baby, bottle, bowl, can,
chimpanzee, clock, cup, fox, keyboard, lion,
orange, pear, plate, snail, spider,
sweet_pepper, telephone, television,
willow_tree, worm

True label: girl

APS set: baby, beetle, bicycle, boy, cockroach,
girl, hamster, motorcycle, mouse, seal

EPIC set: baby, beetle, bicycle, boy, cockroach,
girl, hamster, motorcycle, mouse, seal,
trout

True label: pear

APS set: caterpillar, crocodile, cup, lizard,
seal, snail, snake, spider, worm

EPIC set: caterpillar, crocodile, cup, lizard,
pear, plate, seal, snail, snake,
spider, worm

Figure 4: Additional outlier image prediction sets examples. EPICSCORE consistently produces broader prediction sets for all selected outlier images, effectively capturing the high epistemic uncertainty associated with these observations.

True label: television

APS set: bed, butterfly, can, lobster, motorcycle, streetcar, sunflower, **television**, tractor, wardrobe

EPIC set: can, motorcycle, streetcar, **television**, tractor, wardrobe

True label: apple

APS set:

EPIC set: **apple**, orange

True label: telephone

APS set: clock, keyboard, plate, **telephone**

EPIC set: keyboard, plate, **telephone**

True label: keyboard

APS set: **keyboard**, man, plain, plate, skyscraper, streetcar, telephone, woman

EPIC set: **keyboard**, streetcar, telephone

Figure 5: Additional inlier prediction sets examples. `EPICSCORE` generates more compact prediction sets for all selected inliers while also preventing empty sets in one instance. This highlights its robustness and reliability in regions with low epistemic uncertainty.

B PREDICTIVE DISTRIBUTIONS VIA MONTE CARLO DROPOUT AND BATCH NORMALIZATION

Monte Carlo (MC) dropout was originally introduced as a regularization technique to prevent neural network overfitting [Srivastava et al., 2014]. However, it can also be used to approximate the predictive distribution by interpreting dropout as a variational Bayesian method. Indeed, Gal and Ghahramani [2016] demonstrated that applying dropout at both training and prediction corresponds to a variational approximation to the posterior over the network weights. Specifically, the dropout masks define a variational family where each mask corresponds to a Bernoulli-distributed perturbation of the network. From a practical standpoint, the predictive distribution is obtained by performing multiple stochastic forward passes with dropout enabled at prediction time [Izbicki, 2025]. Each pass applies a different dropout mask, resulting in varied outputs—in our case, different sets of means, variances, and weights of a Gaussian mixture for the same data point $\mathbf{x}$. A sample of Y is then drawn from each configuration, producing a Monte Carlo sample from the predictive distribution $Y|\mathbf{x}, D$.

Batch normalization (BN) [Ioffe and Szegedy, 2015] also introduces stochasticity, but through its reliance on mini-batch statistics during training. As shown by Teye et al. [2018], this stochasticity induces an implicit distribution over the network's parameters, which can be interpreted as a form of approximate Bayesian prediction. During prediction, the batch statistics are fixed, and the model's outputs can be treated as samples from an approximate posterior—similarly to the MC dropout case.
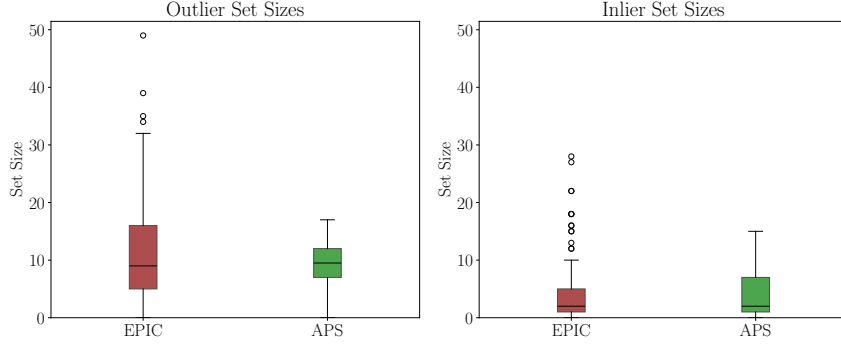
Figure 6: Left: Prediction set sizes for the top 150 most outlying observations. Right: Prediction set sizes for the top 150 most typical (inlier) observations. Both methods consistently produce larger prediction sets for outliers compared to inliers, but `EPICSCORE` shows a more dispersed boxplot with higher set sizes for outliers, and a more concentrated boxplot for inliers compared to APS.

C ADDITIONAL RESULTS AND DETAILS FOR REAL DATA EXPERIMENTS

In this section, we provide an overview of the evaluation metrics, dataset summaries, and additional results for both quantile and standard regression experiments. Additionally, we outline the architecture and hyperparameter configurations for each base model.

C.1 EVALUATION METRICS

Let $\hat{R}(\cdot)$ denote a generic prediction interval. Given a test set $(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_m, Y_m)$, we evaluate performance using the following metrics:

- **Average Marginal Coverage (AMC)**:

$$\text{AMC} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left(Y_i \in \hat{R}(\mathbf{X}_i)\right),$$

  which is an estimate of the marginal coverage of R.

- **Average Interval Score Loss (AISL)** [Gneiting and Raftery, 2007]:

$$\text{AISL} = \frac{1}{m} \sum_{i=1}^{m} \left[\left(\max \hat{R}(\mathbf{X}_i) - \min \hat{R}(\mathbf{X}_i)\right) \right.$$
$$+ \frac{2}{\alpha} \cdot \left(\min \hat{R}(\mathbf{X}_i) - Y_i\right) \cdot \mathbb{I}\left\{Y_i < \min \hat{R}(\mathbf{X}_i)\right\}$$
$$\left. + \frac{2}{\alpha} \cdot \left(Y_i - \max \hat{R}(\mathbf{X}_i)\right) \cdot \mathbb{I}\left\{Y_i > \max \hat{R}(\mathbf{X}_i\right\} \right],$$

  where $\min \hat{R}(\mathbf{X})$ and $\max \hat{R}(\mathbf{X})$ represent the lower and upper bounds of the prediction interval, respectively, and α is the miscalibration level. The Interval Score Loss balances two key objectives: maintaining narrow prediction intervals while penalizing those that fail to cover Y_i, with larger penalties for greater deviations. By averaging these scores across all instances, AISL provides a measure that prioritizes the shortest interval while ensuring sufficient coverage. We chose the AISL following prior work in the literature, where it is commonly used as a summary metric [Rossellini et al., 2024], a paper that also addresses epistemic uncertainty in conformal predictions.

- **Interval Length (IL)**:

$$\text{IL} = \frac{1}{m} \sum_{i=1}^{m} \max \hat{R}(\mathbf{X}_i) - \min \hat{R}(\mathbf{X}_i),$$

which measures the average interval length, reflecting the precision of the predictive intervals. Larger values correspond to wider, less informative intervals, while smaller values indicate more compact and precise intervals.

- **Pearson Correlation between Coverage and Interval Length** (ρ) [Feldman et al., 2021]: This metric measures the correlation between the width of the prediction interval and its coverage, providing insight into potential conditional coverage violations. Specifically,

$$\rho = \left| \frac{\text{Cov}(\mathbf{C}, \mathbf{W})}{\sigma_{\mathbf{C}} \sigma_{\mathbf{W}}} \right|,$$

where $\mathbf{C} = (C_1, \ldots, C_m)$ represents a binary vector, with $C_i = \mathbb{I}(Y_i \in \hat{R}(\mathbf{X}_i))$ indicating whether the prediction interval $\hat{R}(\mathbf{X}_i)$ covers Y_i, and $\mathbf{W} = (W_1, \ldots, W_m)$, where $W_i = \max \hat{R}(\mathbf{X}_i) - \min \hat{R}(\mathbf{X}_i)$. A strong correlation between coverage and interval width suggests a potential violation of conditional coverage, which requires their independence [Feldman et al., 2021]. However, $\rho = 0$ does not guarantee conditional coverage, as non-adaptive methods like regression split can achieve zero correlation by maintaining constant-width intervals [Rossellini et al., 2024]. Thus, while this metric provides a useful proxy for assessing conditional coverage, it is not a definitive measure.

- **Outlier-to-inlier interval length ratio (ILR)**:

$$\text{ILR} = \frac{\frac{1}{|I_{\text{out}}|} \sum_{i \in I_{\text{out}}} \max \hat{R}(\mathbf{X}_i) - \min \hat{R}(\mathbf{X}_i)}{\frac{1}{|I_{\text{in}}|} \sum_{i \in I_{\text{in}}} \max \hat{R}(\mathbf{X}_i) - \min \hat{R}(\mathbf{X}_i)},$$

where I_{out} and I_{in} denote the sets of outlier and inlier indices in the test set, respectively. Following the procedure described in Section A.2, we distinguish inliers from outliers by first reducing dimensionality with t-SNE [Van der Maaten and Hinton, 2008], and then applying the Local Outlier Factor (LOF) method [Breunig et al., 2000] using a contamination rate of 5% (i.e., treating 5% of the test instances as outliers). For this metric, we include all detected outliers in I_{out} and define inliers I_{in} as the 15% of instances with the lowest LOF scores. This metric captures how adaptively a method responds to epistemic uncertainty: higher ILR values indicate that outliers—typically found in data-sparse regions—are assigned wider prediction intervals relative to inliers, as expected from a well-calibrated uncertainty-aware method.

- **Average coverage on outliers (ACO)**:

$$\text{ACO} = \frac{1}{|I_{\text{out}}|} \sum_{i \in I_{\text{out}}} \mathbb{I}(Y_i \in \hat{R}(\mathbf{X}_i)),$$

where I_{out} is the set of outlier indices in the test set, obtained using the same outlier detection procedure described above. This metric measures the proportion of outlier instances whose true response Y_i falls within the predtion interval $\hat{R}(\cdot)$. Higher ACO values—ideally close to the nominal level $1 - \alpha$ indicate that the method maintains reliable coverage even in data-sparse regions. As such, ACO complements the ILR metric by evaluating whether the adaptively wider intervals assigned to outliers are indeed well-calibrated.

C.2 ADDITIONAL RESULTS

Table 5 summarizes the dataset details. Quantile regression results are reported in Tables 6, 7, 8, and 9, while standard regression results appear in Tables 10, 11, and 12. Additional metrics on outlier-to-inlier interval length ratios and outlier coverage are presented in Tables 9 (quantile regression), 13, and 14 (regression).

Table 5: Summary of the datasets used in this paper, including the number of samples (n), features (p), and access links.

Dataset	n	p	Source	Dataset	n	p	Source
Airfoil	1503	5	Airfoil (UCI)	Protein	45730	8	Protein (UCI)
Bike	10885	12	Bike (Kaggle)	Star	2161	48	Star (Harvard Dataverse)
Concrete	1030	8	Concrete (UCI)	SuperConductivity	21263	81	Superconductivity (UCI)
Cycle	9568	4	Cycle (UCI)	Wave Energy Converter	54000	49	WEC (UCI)
Homes	21613	17	Home (Kaggle)	Winered	4898	11	Wine red (UCI)
Eletric	10000	12	Electric (UCI)	WineWhite	1599	11	Wine white (UCI)
Meps19	15781	141	Meps19 (AHRQ site))				

Table 6: Quantile regression Mean Average Coverage values across different methods and datasets. The reported values represent the average over 50 runs, with two times the standard deviation in parentheses. As expected for conformal methods, all approaches achieve marginal coverage close to the nominal level of 0.9

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	CQR	CQR-r	UACQR-P	UACQR-S
airfoil	0.9 (0.008)	0.9 (0.01)	0.896 (0.01)	0.901 (0.007)	0.901 (0.007)	0.907 (0.009)	0.9 (0.007)
bike	0.9 (0.003)	0.898 (0.003)	0.899 (0.003)	0.899 (0.002)	0.899 (0.002)	0.9 (0.002)	0.9 (0.002)
concrete	0.905 (0.008)	0.9 (0.009)	0.898 (0.01)	0.897 (0.007)	0.897 (0.007)	0.914 (0.012)	0.895 (0.007)
cycle	0.899 (0.003)	0.9 (0.003)	0.898 (0.004)	0.901 (0.003)	0.902 (0.003)	0.901 (0.002)	0.9 (0.002)
electric	0.9 (0.003)	0.901 (0.003)	0.902 (0.004)	0.901 (0.002)	0.901 (0.002)	0.901 (0.002)	0.901 (0.002)
homes	0.902 (0.003)	0.902 (0.003)	0.9 (0.003)	0.901 (0.002)	0.901 (0.002)	0.901 (0.002)	0.901 (0.002)
meps19	0.9 (0.003)	0.9 (0.003)	0.9 (0.003)	0.899 (0.002)	0.899 (0.002)	0.901 (0.002)	0.899 (0.002)
protein	0.897 (0.003)	0.897 (0.003)	0.897 (0.003)	0.9 (0.001)	0.9 (0.001)	0.901 (0.001)	0.9 (0.001)
star	0.902 (0.006)	0.902 (0.006)	0.903 (0.006)	0.902 (0.004)	0.901 (0.004)	0.93 (0.013)	0.902 (0.004)
superconductivity	0.898 (0.004)	0.898 (0.003)	0.898 (0.003)	0.9 (0.002)	0.9 (0.002)	0.9 (0.001)	0.9 (0.002)
WEC	0.897 (0.003)	0.899 (0.003)	0.897 (0.004)	0.9 (0.001)	0.9 (0.001)	0.899 (0.001)	0.9 (0.001)
winered	0.906 (0.008)	0.905 (0.008)	0.904 (0.007)	0.897 (0.006)	0.897 (0.006)	0.903 (0.009)	0.897 (0.006)
winewhite	0.901 (0.004)	0.9 (0.005)	0.9 (0.004)	0.898 (0.003)	0.898 (0.003)	0.908 (0.009)	0.898 (0.003)

Table 7: Quantile regression Interval Length values across different methods and datasets. The reported values represent the average over 50 runs, with two times the standard deviation in parentheses. Bolded values indicate the best-performing method within a 95% confidence interval. Overall, EPICSCORE consistently produces narrower intervals in most cases.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	CQR	CQR-r	UACQR-P	UACQR-S
airfoil	16.521 (0.18)	**16.395 (0.237)**	**16.02 (0.222)**	17.087 (0.127)	17.09 (0.124)	18.838 (0.367)	16.656 (0.386)
bike $\times(10^1)$	37.042 (0.190)	38.413 (0.229)	**36.250 (0.218)**	41.164 (0.150)	41.125 (0.153)	43.627 (0.591)	37.415 (0.386)
concrete	**36.537 (0.375)**	38.328 (0.618)	37.614 (0.651)	39.477 (0.353)	39.486 (0.36)	44.425 (1.307)	39.853 (1.536)
cycle	**30.975 (0.128)**	31.587 (0.132)	**30.714 (0.146)**	35.235 (0.095)	35.346 (0.093)	38.292 (0.195)	31.045 (0.207)
electric	0.088 (0.001)	0.084 (0.001)	**0.072 (0.001)**	0.09 (0.001)	0.09 (0.001)	0.097 (0.001)	0.084 (0.001)
homes $\times(10^5)$	5.888 (0.028)	5.739 (0.028)	5.816 (0.040)	6.313 (0.024)	6.259 (0.024)	6.750 (0.0302)	**5.312 (0.0309)**
meps19	32.996 (0.314)	29.268 (0.262)	29.16 (0.266)	28.948 (0.249)	28.949 (0.249)	**27.857 (0.314)**	32.763 (0.815)
protein	16.195 (0.039)	16.485 (0.052)	**16.048 (0.034)**	16.378 (0.011)	16.378 (0.011)	16.797 (0.019)	16.356 (0.017)
star $\times(10^1)$	**81.851 (1.029)**	81.760 (1.042)	82.050 (1.083)	81.359 (0.508)	81.396 (0.5117)	82.521 (0.618)	83.253 (0.952)
superconductivity	66.906 (0.205)	70.631 (0.366)	**64.805 (0.197)**	69.51 (0.144)	69.482 (0.145)	78.478 (0.7)	67.046 (0.492)
WEC $\times(10^5)$	2.401 (0.0111)	2.076 (0.011)	**1.890 (0.009)**	2.708 (0.004)	2.709 (0.004)	2.843 (0.008)	2.547 (0.007)
winered	2.098 (0.034)	2.096 (0.035)	2.11 (0.031)	**1.906 (0.011)**	**1.902 (0.01)**	2.031 (0.025)	2.077 (0.042)
winewhite	2.31 (0.017)	2.212 (0.029)	2.253 (0.016)	**2.12 (0.006)**	2.121 (0.006)	**2.124 (0.012)**	2.222 (0.017)

C.3 BASE MODEL HYPERPARAMETERS

For the CatBoost quantile regression model [Dorogush et al., 2018], we set the number of iterations (trees) to 1,000 and the learning rate to 0.001, enabling early stopping after 50 rounds of no improvement. To mitigate overfitting even further, we limit each tree to a maximum depth of 6. All other parameters follow the default CatBoost settings. In the regression setting, we implemented a Neural Network with three hidden layers, consisting of 64, 32, and 16 neurons, respectively. Each layer utilizes ReLU activation, batch normalization [Ioffe and Szegedy, 2015], and dropout rates [Srivastava et al., 2014] of 0.2, 0.1, and 0.05, correspondingly. We train the model using a smooth L1 loss, as it provides a balance between mean absolute error (MAE) and mean squared error (MSE), making it more robust to outliers while maintaining stable gradient updates.

For optimization, we utilize the Adam optimizer [Kingma, 2014] with an initial learning rate of 0.01. A learning rate scheduler is incorporated to decrease the learning rate by a factor of 0.5 if there is no improvement after 10 epochs, accelerating convergence. All weights are initialized using Xavier normal initialization [Kumar, 2017]. We set aside 30% of the training data for validation and set a batch size of 35. Training proceeds for a maximum of 750 epochs, with early stopping triggered if there is no improvement on the validation set for 30 consecutive epochs. Additionally, feature scaling and target min-max normalization are applied to ensure stable training.

D EPICSCORE COMPUTATIONAL AND METHODOLOGICAL DETAILS

This section outlines the specifications of each predictive model used in our framework, discusses the scalability of each approach, and details the trade-offs associated with prior choices.

Table 8: Quantile regression Pearson correlation values across different methods and datasets. The reported values represent the average over 50 runs, with two times the standard deviation in parentheses. Bolded values indicate the best-performing method within a 95% confidence interval. Overall, EPICSCORE exhibits low correlation in most cases, reflecting strong conditional coverage performance.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	CQR	CQR-r	UACQR-P	UACQR-S
airfoil	**0.06 (0.013)**	0.18 (0.016)	**0.071 (0.016)**	0.125 (0.017)	0.129 (0.017)	0.132 (0.033)	0.108 (0.02)
bike	0.171 (0.013)	0.138 (0.013)	0.213 (0.012)	**0.062 (0.013)**	**0.069 (0.013)**	0.108 (0.016)	0.091 (0.012)
concrete	0.101 (0.021)	0.147 (0.02)	**0.068 (0.013)**	**0.081 (0.017)**	**0.082 (0.017)**	0.121 (0.034)	**0.088 (0.02)**
cycle	**0.046 (0.008)**	**0.045 (0.009)**	0.085 (0.01)	0.27 (0.011)	0.292 (0.011)	0.255 (0.011)	0.192 (0.012)
electric	**0.023 (0.006)**	**0.044 (0.009)**	0.159 (0.008)	0.075 (0.006)	0.071 (0.006)	0.123 (0.01)	0.134 (0.008)
homes	0.122 (0.007)	0.143 (0.01)	**0.048 (0.008)**	0.126 (0.008)	0.15 (0.007)	0.271 (0.008)	0.221 (0.007)
meps19	**0.017 (0.004)**	0.084 (0.01)	0.069 (0.009)	0.08 (0.006)	0.08 (0.006)	0.128 (0.006)	0.051 (0.006)
protein	**0.031 (0.004)**	0.15 (0.01)	0.073 (0.004)	0.094 (0.003)	0.094 (0.003)	0.116 (0.004)	0.094 (0.004)
star	0.07 (0.012)	**0.042 (0.009)**	0.085 (0.012)	**0.046 (0.009)**	**0.047 (0.01)**	**0.048 (0.011)**	**0.041 (0.009)**
superconductivity	0.167 (0.006)	0.217 (0.008)	**0.034 (0.007)**	0.069 (0.007)	0.073 (0.006)	0.137 (0.008)	0.088 (0.005)
WEC	**0.122 (0.004)**	**0.119 (0.006)**	0.149 (0.005)	0.136 (0.005)	0.147 (0.005)	**0.13 (0.007)**	0.132 (0.005)
winered	**0.062 (0.013)**	**0.085 (0.016)**	**0.058 (0.011)**	0.113 (0.015)	0.114 (0.016)	0.097 (0.019)	**0.076 (0.016)**
winewhite	**0.068 (0.016)**	0.13 (0.016)	**0.072 (0.012)**	0.147 (0.011)	0.147 (0.011)	0.156 (0.014)	0.099 (0.011)

Table 9: Average coverage on outliers across methods and datasets in the quantile regression setting. This table reports the average prediction interval coverage for outlier observations—identified as data-sparse points—across 50 runs. Bold values indicate the best-performing method within a 95% confidence interval. EPICSCORE consistently delivers near-nominal coverage across most datasets.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	CQR	CQR-r	UACQR-P	UACQR-S
airfoil	**0.894 (0.029)**	**0.875 (0.032)**	**0.902 (0.024)**	**0.877 (0.031)**	**0.876 (0.032)**	**0.875 (0.033)**	**0.872 (0.031)**
bike	**0.885 (0.009)**	**0.9 (0.011)**	**0.894 (0.011)**	**0.882 (0.01)**	**0.882 (0.01)**	**0.887 (0.009)**	**0.885 (0.009)**
concrete	**0.824 (0.039)**	**0.855 (0.034)**	**0.875 (0.038)**	**0.845 (0.035)**	**0.844 (0.036)**	**0.878 (0.032)**	**0.845 (0.034)**
cycle	**0.907 (0.009)**	**0.908 (0.01)**	**0.906 (0.011)**	**0.9 (0.009)**	**0.901 (0.009)**	**0.909 (0.01)**	**0.906 (0.009)**
electric	**0.887 (0.01)**	**0.887 (0.009)**	**0.895 (0.009)**	**0.888 (0.009)**	**0.888 (0.009)**	**0.889 (0.01)**	**0.889 (0.009)**
homes	**0.89 (0.007)**	**0.885 (0.007)**	**0.895 (0.008)**	**0.883 (0.007)**	**0.88 (0.007)**	0.864 (0.009)	0.872 (0.007)
meps19	**0.904 (0.008)**	**0.903 (0.008)**	**0.899 (0.009)**	**0.897 (0.008)**	**0.897 (0.008)**	**0.9 (0.008)**	**0.897 (0.008)**
protein	**0.893 (0.005)**	**0.895 (0.005)**	**0.896 (0.005)**	**0.894 (0.004)**	**0.894 (0.004)**	**0.894 (0.005)**	**0.894 (0.004)**
star	0.867 (0.016)	0.871 (0.017)	**0.876 (0.016)**	**0.875 (0.018)**	**0.875 (0.018)**	**0.911 (0.021)**	0.87 (0.018)
superconductivity	0.863 (0.008)	0.872 (0.009)	0.879 (0.008)	**0.913 (0.006)**	**0.912 (0.006)**	0.932 (0.005)	**0.912 (0.006)**
WEC	**0.901 (0.006)**	0.864 (0.008)	0.86 (0.009)	0.927 (0.005)	0.926 (0.005)	0.917 (0.005)	0.921 (0.005)
winered	**0.875 (0.029)**	**0.876 (0.03)**	**0.892 (0.026)**	0.846 (0.032)	0.847 (0.032)	0.861 (0.033)	0.848 (0.032)
winewhite	**0.86 (0.015)**	**0.854 (0.016)**	**0.868 (0.014)**	0.847 (0.016)	0.847 (0.016)	0.865 (0.02)	0.861 (0.016)

D.1 SPLITTING STRATEGY

When splitting the calibration set $\mathcal{D}_{cal}$ for fitting the predictive model and deriving EPICSCORE's adaptive cutoffs in disjunct data subsets, we prioritize allocating the majority of data to model training, as cutoff derivation primarily involves a simpler quantile computation. Specifically, for small with $n \leq 3000$, we use 30% of the calibration samples for cutoff computation. For larger datasets, this allocation is capped at 1000 samples to maintain computational efficiency. This approach ensures that EPICSCORE achieves both accurate predictive distribution estimates and well-calibrated cutoffs.

D.2 MDN MC-DROPOUT DETAILS

The employed Mixture Density Network (MDN) consists of 2 hidden layers, each with 64 neurons, ReLU activations, and batch normalization [Ioffe and Szegedy, 2015] to enhance training stability. The output layer predicts three parameters for each mixture component $k = \{1, \ldots, K\}$: membership probability $\pi_k(\cdot)$, the mixture mean $\mu_k(\cdot)$ and the mixture variance $\sigma_k^2(\cdot)$. We set the number of components to $K = 3$, outputting 9 parameters. The network is trained using the negative log-likelihood loss, applying a softmax activation to the mixture probabilities and a softplus activation to the variance estimates.

Optimization relies on the Adam optimizer [Kingma, 2014] with a learning rate of 0.001 and a step learning rate scheduler that decays by 0.99 every 5 epochs. To model epistemic uncertainty, we incorporate MC dropout [Gal and Ghahramani, 2016], applying a dropout rate of 0.5 to each hidden layer. This setup implicitly corresponds to a diffuse, non-informative

Table 10: Regression Mean Average Coverage values across different methods and datasets. The reported values represent the average over 50 runs, with two times the standard deviation in parentheses. As expected for conformal methods, all approaches maintain marginal coverage close to the nominal level of 0.9.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	Mondrian	Reg-split	Weighted
airfoil	0.897 (0.01)	0.9 (0.008)	0.897 (0.009)	0.906 (0.006)	0.897 (0.007)	0.9 (0.007)
bike	0.901 (0.003)	0.903 (0.003)	0.898 (0.003)	0.904 (0.002)	0.899 (0.002)	0.9 (0.002)
concrete	0.897 (0.009)	0.902 (0.011)	0.907 (0.009)	0.929 (0.006)	0.901 (0.008)	0.896 (0.008)
cycle	0.898 (0.004)	0.9 (0.004)	0.896 (0.004)	0.905 (0.003)	0.898 (0.003)	0.9 (0.002)
electric	0.899 (0.003)	0.9 (0.003)	0.896 (0.003)	0.905 (0.002)	0.899 (0.003)	0.901 (0.003)
homes	0.9 (0.003)	0.899 (0.003)	0.9 (0.004)	0.902 (0.002)	0.901 (0.002)	0.9 (0.002)
meps19	0.899 (0.003)	0.9 (0.003)	0.897 (0.003)	0.902 (0.006)	0.9 (0.002)	0.9 (0.002)
protein	0.9 (0.003)	0.901 (0.003)	0.899 (0.002)	0.9 (0.001)	0.9 (0.001)	0.899 (0.001)
star	0.903 (0.005)	0.9 (0.006)	0.906 (0.006)	0.913 (0.005)	0.903 (0.004)	0.9 (0.004)
superconductivity	0.901 (0.003)	0.901 (0.003)	0.9 (0.004)	0.901 (0.002)	0.899 (0.002)	0.899 (0.002)
WEC	0.901 (0.002)	0.898 (0.003)	0.899 (0.003)	0.9 (0.001)	0.899 (0.001)	0.9 (0.001)
winered	0.898 (0.009)	0.9 (0.008)	0.895 (0.008)	0.91 (0.005)	0.903 (0.006)	0.895 (0.005)
winewhite	0.902 (0.004)	0.904 (0.004)	0.901 (0.004)	0.911 (0.003)	0.9 (0.004)	0.899 (0.003)

Table 11: Regression Interval length values across different methods and datasets. The reported values represent the average over 50 runs, with two times the standard deviation in parentheses. Bold values indicate the best-performing method within a 95% confidence interval. In general, our framework produces narrower intervals in most datasets.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	Mondrian	Reg-split	Weighted
airfoil	**15.099 (0.592)**	**15.223 (0.591)**	**15.089 (0.56)**	16.671 (0.632)	**15.325 (0.469)**	**15.693 (0.521)**
bike $\times(10^1)$	**24.910 (0.302)**	26.616 (0.333)	25.665 (0.338)	27.263 (0.264)	27.634 (0.271)	25.865 (0.249)
concrete	**39.025 (1.49)**	**39.44 (1.575)**	**40.475 (1.551)**	51.284 (1.904)	**39.943 (1.204)**	43.053 (1.629)
cycle	**14.911 (0.183)**	**14.851 (0.199)**	**14.712 (0.174)**	15.015 (0.164)	**14.855 (0.159)**	**14.833 (0.15)**
electric	**0.036 (<0.001)**	0.037 (<0.001)	**0.036 (<0.001)**	0.038 (<0.001)	0.037 (<0.001)	0.037 (<0.001)
homes $\times(10^5)$	**3.848 (0.043)**	**3.758 (0.049)**	4.040 (0.059)	4.235 (0.037)	4.014 (0.033)	3.984 (0.032)
meps19	**25.013 (0.87)**	**26.605 (0.774)**	32.093 (1.446)	38.904 (1.039)	28.899 (0.544)	29.555 (0.843)
protein	14.572 (0.106)	14.311 (0.12)	**13.573 (0.103)**	14.102 (0.038)	15.261 (0.037)	**13.692 (0.037)**
star $\times(10^1)$	**85.3 (1.015)**	85.148 (1.326)	86.499 (1.288)	90.539 (0.762)	**85.230 (0.792)**	104.202 (1.306)
superconductivity	**39.13 (0.419)**	39.283 (0.479)	**39.547 (0.5)**	42.204 (0.216)	46.14 (0.242)	40.115 (0.228)
WEC $\times(10^5)$	0.893 (0.011)	**0.858(0.011)**	0.900 (0.012)	0.903 (0.004)	0.925 (0.006)	0.879 (0.006)
winered	**2.361 (0.065)**	**2.37 (0.067)**	**2.316 (0.054)**	2.576 (0.04)	**2.39 (0.037)**	2.541 (0.051)
winewhite	**2.337 (0.032)**	**2.4 (0.032)**	**2.356 (0.031)**	2.445 (0.013)	2.361 (0.014)	2.387 (0.015)

prior over the network weights—one that does not favor any specific parameter configuration and assumes equal uncertainty over weights. All network weights are initialized using PyTorch's default scheme. To monitor generalization, 30% of the training data is held out for validation. The model is trained for up to 2,000 epochs, with early stopping after 50 epochs without validation improvement. Feature scaling and target normalization are also applied to improve numerical stability and parameter estimation.

Batch size selection is dataset-dependent: 40 for small datasets ($n < 10000$), 125 for medium datasets ($n < 50000$), and 250 for large-scaled datasets, such as WEC. For the image experiment, we introduce an additional hidden layer with 32 neurons and maintain a dropout rate of 0.5, while adjusting the batch size to 135. To compute the predictive CDF $F(s(\mathbf{X}, Y)|\mathbf{X}, D)$ at $\mathbf{X} = \mathbf{x}$, we generate 500 samples of the mixture parameters using MC dropout forward passes. For each sampled mixture parameter set, score samples are drawn from the Gaussian Mixture Model. The final predictive CDF is obtained by computing the empirical distribution of $s(\mathbf{X}, Y)$ over these score samples.

D.3 VARIATIONAL GP DETAILS

For the Variational Gaussian Process (GP), we implement the model using the *gpytorch* package [Gardner et al., 2018] and PyTorch. The GP prior is specified with a constant mean function and a Radial Basis Function (RBF) kernel, which provides a smoothness assumption but remains essentially non-informative in the sense that it does not encode strong prior beliefs about the function values or parameters. To approximate the posterior, we employ a Natural Variational Distribution [Salimbeni et al., 2018], using 15 inducing points for small datasets ($n < 10000$) and 50 for medium and large-scale data.

Table 12: Regression Pearson correlation values across different methods and datasets. The reported values represent the average over 50 runs, with two times the standard deviation in parentheses. Bold values indicate the best-performing method within a 95% confidence interval. The Pearson correlation for Regression Split is omitted, as its constant interval length results in an undefined correlation value. Overall, EPICSCORE achieves low correlations in most cases, indicating strong conditional coverage performance.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	Mondrian	Weighted
airfoil	**0.056 (0.013)**	0.125 (0.018)	**0.054 (0.012)**	0.148 (0.017)	0.124 (0.016)
bike	0.164 (0.009)	0.172 (0.006)	0.054 (0.007)	**0.028 (0.005)**	0.043 (0.007)
concrete	**0.064 (0.015)**	0.116 (0.019)	**0.054 (0.011)**	0.191 (0.02)	0.211 (0.022)
cycle	**0.022 (0.005)**	0.075 (0.008)	**0.023 (0.005)**	0.043 (0.006)	**0.025 (0.005)**
electric	0.052 (0.007)	0.128 (0.009)	**0.024 (0.005)**	0.047 (0.007)	**0.029 (0.006)**
homes	0.135 (0.007)	**0.19 (0.011)**	0.019 (0.005)	**0.016 (0.003)**	0.038 (0.005)
meps19	0.17 (0.012)	0.183 (0.015)	**0.034 (0.008)**	**0.022 (0.006)**	0.053 (0.016)
protein	0.063 (0.003)	0.071 (0.005)	0.062 (0.003)	**0.013 (0.003)**	0.043 (0.005)
star	0.076 (0.012)	**0.037 (0.01)**	0.073 (0.01)	0.156 (0.012)	0.335 (0.016)
superconductivity	0.072 (0.006)	0.254 (0.005)	**0.016 (0.004)**	**0.019 (0.004)**	**0.025 (0.006)**
WEC	**0.012 (0.002)**	0.115 (0.004)	0.21 (0.007)	**0.009 (0.002)**	0.059 (0.007)
winered	**0.05 (0.01)**	0.119 (0.016)	**0.042 (0.009)**	0.153 (0.018)	0.221 (0.019)
winewhite	**0.035 (0.007)**	0.079 (0.011)	**0.025 (0.005)**	0.055 (0.009)	0.092 (0.011)

Training is performed by minimizing the negative variational ELBO, where the Natural Gradient Descent [Salimbeni et al., 2018] updates the variational parameters, while the Adam optimizer [Kingma, 2014] refines the GP kernel and noise variance hyperparameters.

The model is trained for up to 2000 epochs, with early stopping triggered after 50 epochs of no improvement. Following the MDN predictive model, we reserve 30% of the training data exclusively for validation and adopt an adaptive batch size of 40 for small datasets ($n < 10000$), 125 for medium-sized datasets ($n < 50000$), and 250 for large datasets. To ensure numerical stability, we apply feature scaling and target normalization. Using the learned variational gaussian posterior, we easily derive the predictive CDF $F(s(\mathbf{X}, Y)|\mathbf{X}, D)$ at $\mathbf{X} = \mathbf{x}$ by using gaussian conjugacy.

D.4 BART DETAILS

For the Bayesian Additive Regression Tree (BART) model, we use the base implementation from *pymc3* [Quiroga et al., 2022] and adopt the heteroscedastic variant [Pratola et al., 2020]. The conformal scores are modeled as a normal distribution, where the mean is determined by the sum of regression trees, and the variance depends on $\mathbf{X}$. We configure the model with 100 trees and retain the default prior settings provided by *pymc3*, which correspond to a non-informative prior favoring deeper trees. This choice allows for flexible modeling without imposing strong structural constraints. Additionally, we apply target normalization to improve numerical stability and enhance posterior inference.

After obtaining BART posterior samples via MCMC [Chipman et al., 2012], we derive the predictive CDF $F(S(\mathbf{X}, Y)|\mathbf{X}, D)$ for a given $\mathbf{X} = \mathbf{x}$ by simulating scores from the postulated distribution using the posterior samples. The corresponding empirical CDF for the score $S(\mathbf{X}, Y)$ is then computed. In the regression example presented in Figure 1 and detailed in Appendix A.1, we used a modified version of BART. In this version, a heteroscedastic gamma distribution was assigned to the score instead of a normal distribution, which was necessary to account for the asymmetry in the regression conformal scores for that example.

D.5 IMPACT OF PRIOR CHOICES

In this section, we illustrate the impact of prior choices in EPICSCORE through a simulated experiment comparing the prediction intervals produced by EPICSCORE-BART under two prior configurations: *diffuse* and *concentrated*. To emulate data sparsity, we follow a similar setup to Section A.1, simulating a scenario with both high- and low-density regions. Specifically, given a sample size n, we draw $\lfloor 0.8 \cdot n \rfloor$ samples from $X \sim U(-1, 0)$, representing the denser region. The remaining $\lfloor 0.2 \cdot n \rfloor$ samples are generated from $X \sim \text{Beta}(5, 5)$, concentrating points in the intermediate region of the interval [0,1]. The response variable follows a heteroscedastic distribution $Y \sim N\left(\mathbb{I}\{x > 0\}, 0.05 + (\sin^2(15 \cdot x) \cdot \mathbb{I}\{x > 0\})\right)$, which

Table 13: Average outlier-to-inlier interval length ratio across methods and datasets in the regression setting. The table reports the average ratio between prediction interval lengths for outliers and inliers, averaged over 50 runs. Bold values indicate the best-performing method within a 95% confidence interval. The interval length ration for Regression Split is omitted, as its constant interval length results will always return a ratio of 1. Overall, EPICSCORE consistently yields higher ratios across diverse datasets.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	Mondrian	Weighted
airfoil	**1.021 (0.039)**	**1.004 (0.014)**	**1.03 (0.035)**	**0.988 (0.023)**	**1.03 (0.045)**
bike	0.971 (0.014)	0.98 (0.006)	0.963 (0.02)	**1.029 (0.018)**	0.976 (0.016)
concrete	**1.05 (0.036)**	1.03 (0.018)	**1.11 (0.035)**	0.99 (0.036)	**1.137 (0.067)**
cycle	0.967 (0.007)	0.992 (0.002)	0.973 (0.008)	**1.005 (0.006)**	0.988 (0.005)
electric	0.987 (0.008)	**1.0 (0.003)**	0.992 (0.01)	**0.996 (0.005)**	**0.992 (0.008)**
homes	1.124 (0.019)	1.138 (0.015)	**1.268 (0.035)**	1.234 (0.026)	1.203 (0.035)
meps19	**1.047 (0.027)**	**1.057 (0.041)**	**1.097 (0.065)**	**1.099 (0.05)**	**1.074 (0.058)**
protein	0.999 (0.002)	0.996 (0.003)	**1.008 (0.004)**	**1.003 (0.004)**	**1.0 (0.005)**
star	**1.0 (0.007)**	1.0 (0.001)	**1.005 (0.007)**	**1.016 (0.011)**	0.984 (0.025)
superconductivity	1.129 (0.014)	1.061 (0.009)	1.107 (0.016)	**1.211 (0.014)**	1.111 (0.017)
WEC	1.077 (0.007)	1.034 (0.005)	1.122 (0.015)	**1.2 (0.014)**	1.041 (0.007)
winered	1.055 (0.019)	1.031 (0.009)	**1.143 (0.027)**	1.052 (0.02)	1.095 (0.045)
winewhite	1.01 (0.007)	1.001 (0.001)	**1.039 (0.012)**	**1.058 (0.013)**	0.997 (0.011)

results in low aleatoric and epistemic uncertainty over the densely sampled interval $[-1, 0]$ and high uncertainty—both epistemic and aleatoric—over the sparse interval $[0, 1]$.

The dataset is divided into training, calibration, and test sets, each containing 200 samples. As the base predictor, we use a k-nearest neighbors (k-NN) regression model with $k = 10$, trained on the training set. For EPICSCORE, both BART configurations share the same settings: $m = 50$ trees, heteroscedastic variance, a normal likelihood, response normalization, and 1000 Monte Carlo samples. The key difference lies in the prior hyperparameter β, which governs tree depth by controlling the probability that a node is non-terminal. A higher value, such as $\beta = 0.9$ yields a diffuse prior with deeper trees and greater model flexibility, while a lower value like $\beta = 0.1$ results in a concentrated prior with shallower trees, imposing stronger regularization.

Figure 7 illustrates how the prediction intervals behave under each prior configuration. As shown, the diffuse prior
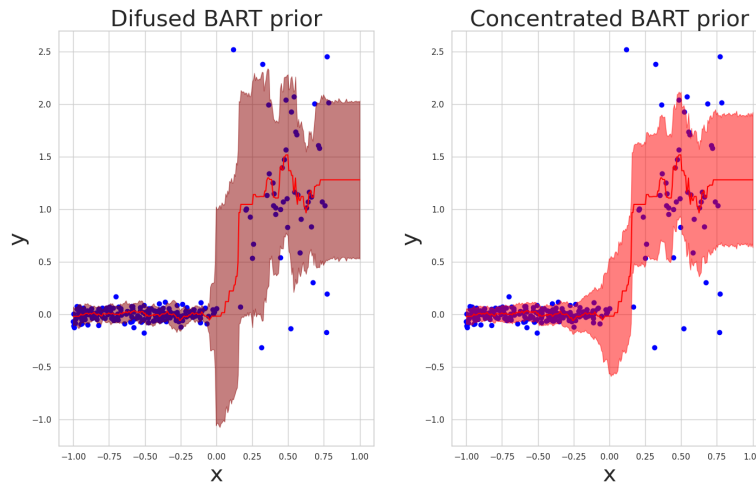

Figure 7: Comparison of prediction regions produced by EPICSCORE using BART under two prior configurations: a diffuse prior and a concentrated prior. While the diffuse prior produces wider prediction regions—particularly in sparse areas—due to greater model flexibility, the concentrated prior yields narrower intervals reflecting stronger regularization. This highlights the sensitivity of EPICSCORE to prior specification.

Table 14: Average coverage on outliers across methods and datasets in the regression setting. This table reports the average prediction interval coverage for outlier observations—identified as data-sparse points—across 50 runs. Coverage values closer to the nominal level of 0.9 reflect better adaptability to epistemic uncertainty. Bold values indicate the best-performing method within a 95% confidence interval. EPICSCORE consistently delivers near-nominal coverage across most datasets.

Dataset	EPIC-BART	EPIC-GP	EPIC-MDN	Mondrian	Reg-split	Weighted
airfoil	**0.896 (0.027)**	0.883 (0.028)	**0.894 (0.027)**	0.91 (0.025)	**0.902 (0.023)**	**0.903 (0.021)**
bike	**0.895 (0.01)**	0.901 (0.011)	**0.897 (0.01)**	0.909 (0.009)	0.903 (0.01)	0.895 (0.009)
concrete	**0.895 (0.029)**	0.902 (0.027)	**0.893 (0.028)**	0.904 (0.023)	0.873 (0.033)	0.876 (0.029)
cycle	**0.89 (0.009)**	0.901 (0.01)	**0.892 (0.008)**	0.914 (0.008)	0.901 (0.009)	0.898 (0.009)
electric	**0.9 (0.008)**	**0.898 (0.01)**	0.902 (0.008)	0.909 (0.008)	0.906 (0.009)	0.911 (0.009)
homes	0.869 (0.008)	0.873 (0.008)	**0.896 (0.006)**	0.894 (0.007)	0.847 (0.008)	0.878 (0.008)
meps19	**0.899 (0.007)**	0.9 (0.007)	**0.899 (0.008)**	0.901 (0.012)	0.894 (0.008)	0.898 (0.007)
protein	**0.89 (0.005)**	0.892 (0.005)	**0.89 (0.005)**	0.898 (0.004)	0.895 (0.004)	0.892 (0.004)
star	**0.892 (0.019)**	0.899 (0.019)	0.906 (0.018)	0.899 (0.015)	0.885 (0.02)	0.879 (0.018)
superconductivity	**0.891 (0.007)**	0.872 (0.008)	**0.887 (0.008)**	0.918 (0.005)	**0.884 (0.007)**	0.888 (0.006)
WEC	0.861 (0.006)	0.85 (0.007)	0.871 (0.006)	**0.897 (0.005)**	0.85 (0.006)	0.852 (0.006)
winered	**0.87 (0.028)**	0.841 (0.029)	0.881 (0.027)	0.858 (0.024)	0.843 (0.031)	0.846 (0.029)
winewhite	0.872 (0.015)	**0.876 (0.014)**	**0.878 (0.016)**	0.9 (0.011)	0.863 (0.015)	0.867 (0.016)

yields wider intervals—particularly in data-sparse regions—capturing heightened epistemic uncertainty. In contrast, the concentrated prior produces tighter intervals in denser regions, reflecting its stronger regularization and more confident assumptions in well-populated areas. This highlights the sensitivity of EPICSCORE to prior specification and demonstrates how prior concentration directly influences the adaptiveness and sharpness of the prediction sets.

To further investigate this trade-off, we also evaluate the influence of β using the AISL metric, which jointly accounts for coverage and interval length. As shown in Figure 8, higher values of β lead to deeper trees and wider intervals, while smaller values yield more concise intervals due to stronger regularization.
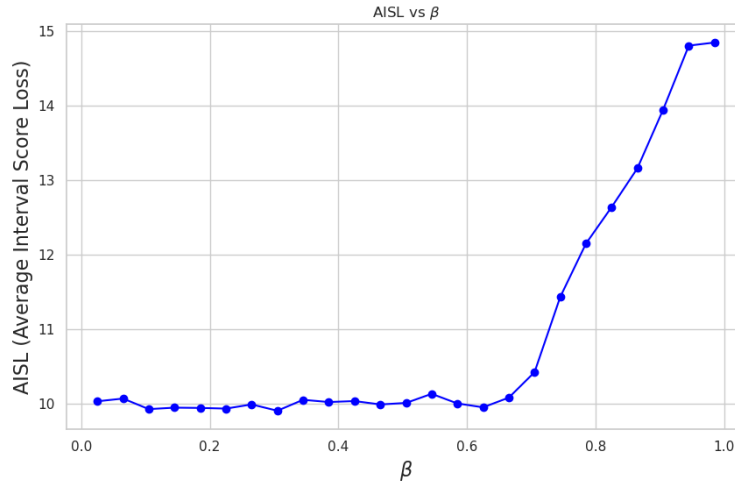

Figure 8: Variation of the **Average Interval Score Loss** (AISL) as a function of the β hyperparameter in the BART model used within the EPICSCORE framework. Lower AISL values indicate more favorable balance, suggesting that tuning β can significantly affect performance by controlling model flexibility and prior regularization.

This analysis helps reveal the trade-off between interval regularization and accurate representation of epistemic uncertainty. Crucially, it also provides a foundation for tuning prior hyperparameters more systematically. By optimizing AISL—or another relevant performance metric—one can select prior configurations in a principled manner. In more complex scenarios involving multiple hyperparameters, this process remains scalable through the use of Bayesian optimization techniques [Frazier, 2018].

D.6 SCALABILITY COMPARISSON

Incorporating Bayesian models into conformal prediction offers a principled way to capture epistemic uncertainty but may introduce additional computational overhead. To quantify this cost within our framework, we evaluate the runtime of `EPICSCORE` across increasing sample sizes using the three Bayesian predictive models discussed previously: MDN, BART, and Variational GP. We consider a synthetic regression task with 30 input features and one response variable, generated using scikit-learn's `make_regression` function [Pedregosa et al., 2011]. For each sample size ($n = 1000, 2000, 5000, 10000, 20000, 50000$), we generate training and calibration sets of equal size (n), train a k-nearest neighbors (k-NN) model with $k = 30$ as the base predictor on the training data, and fit the predictive models on the calibration data. Each setting is repeated 10 times, and Figure 9 reports the average runtime along with its standard deviation for each sample size.
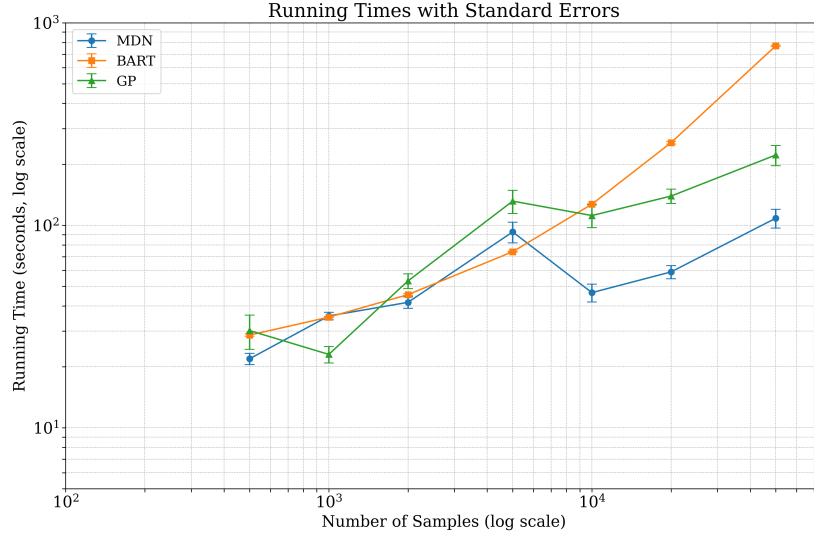

Figure 9: Computational cost of `EPICSCORE` as a function of sample size n (log scale), using the three Bayesian predictive models: BART, MDN and Variational GP. While all three models show comparable running times at moderate scales(e.g., $n = 10000$), MDNs exhibit significantly better scalability and faster execution as n increases, underscoring their advantage in large-scale settings.

The MDN configuration includes two hidden layers (64–64 units), ReLU activations, 50% dropout, 2000 training epochs, a learning rate of 0.001, and batch size dynamically scaled to the sample size (ranging from 35 to 450). Inputs are standardized, and the response is normalized. BART is configured with 50 trees, normal heteroscedastic errors, 500 posterior samples, and parallelization using 6 CPU cores. The GP model is trained via a variational approximation, with the number of inducing points dynamically adjusted from 15 to 150 based on n, a variational learning rate of 0.1, a hyperparameter learning rate of 0.01, and identical batch size and early stopping parameters to the MDN.

Overall, MDNs demonstrate the most stable and scalable runtime across all sample sizes. Variational Gaussian Processes also scale well, though they are generally less efficient than MDNs. The observed drop in runtime between 5000 and 10000 samples for both neural network–based models likely stems from the early stopping mechanism, which halts training once convergence is detected. BART, on the other hand, maintains low computational cost for small datasets but exhibits a sharp increase in runtime as the sample size grows. This highlights a key trade-off: while BART is effective for small to medium-scale problems, neural network–based models like MDNs are better suited for larger datasets, where the iterative nature of MCMC-based methods such as BART becomes increasingly expensive.

This experiment was conducted using an AMD Ryzen 7 6000 series CPU. For the largest dataset, which includes over 100000 calibration observations, the runtime for each method remained small: BART completed in under 16 minutes, MDN in under 2 minutes, and GP in under 4 minutes. These results highlight the computational efficiency of the proposed approaches, particularly the scalability of MDNs and GPs for large-scale applications.

D.7 APPLICATION TO DENSITY-BASED CONFORMITY SCORES

In this section, we use a simulated experiment to illustrate how `EPICSCORE` can enhance density-based conformal prediction methods by explicitly incorporating epistemic uncertainty. Following the same simulation setup described in Section D.5, we compare our approach to a standard method for conformal prediction with a density-based score, namely HPD-split [Izbicki et al., 2022]. The experiment is designed to reflect regions with both high epistemic and aleatoric uncertainty, allowing us to highlight the advantages of our method in capturing complex uncertainty structures.

The base conditional density model is a Mixture Density Network (MDN) consisting of two hidden layers with 64 and 128 neurons, respectively, and a Gaussian mixture output with 5 components. The model incorporates a dropout rate of 0.15 to improve generalization and is trained on 2000 samples. For `EPICSCORE`, we use a Bayesian Additive Regression Trees (BART) model with 50 trees, a gamma error distribution, heteroscedastic variance, and a non-informative prior. The tree depth is controlled by a fixed hyperparameter $\beta = 0.9$, promoting deeper trees. Calibration is performed using 1000 posterior samples. The HPD-split prediction region is constructed directly from the MDN's conditional density estimates. Figure 10 compares the prediction regions obtained by the two methods.
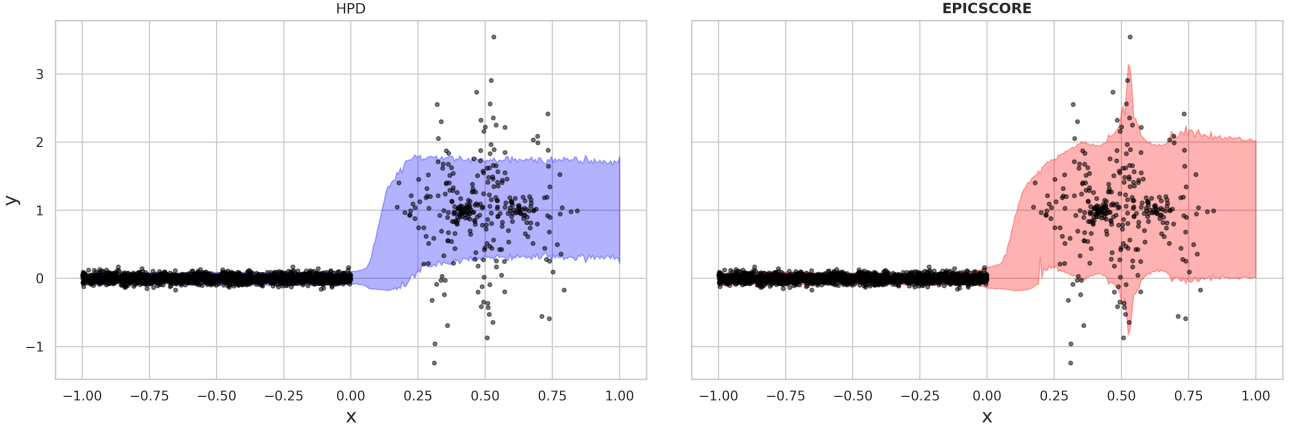

Figure 10: Comparison of prediction regions generated by the HPD (Highest Predictive Density) split [Izbicki et al., 2022] and `EPICSCORE` methods in a simulated scenario. Both methods produce similar regions in the high-density area ($x < 0$), but `EPICSCORE` exhibits sharper adaptivity in the sparse region ($x > 0$), offering more informative and expanded predictive sets where epistemic uncertainty is higher.

Compared to the HPD-based region, our method exhibits significantly better adaptiveness and uncertainty quantification, particularly in data-sparse regions of the input space. At the same time, it preserves tight prediction intervals in denser, data-rich areas. This demonstrates `EPICSCORE`'s ability to more effectively capture uncertainty in challenging settings where data is limited, without sacrificing performance in well-populated regions.

E PROOFS

Proof of Theorem 1. Follows immediately from the fact that `EPICSCORE` is a conformal score and Lei et al. [2018, Theorem 2]. □

Proof of Theorem 2. Let $t'_{1-\alpha}$ be the empirical quantile obtained using $s'(\mathbf{x}, y) = F(s(\mathbf{x}, y)|\mathbf{x}, D)$, and let $t''_{1-\alpha}$ be the empirical quantile obtained using $s''(\mathbf{x}, y) = F(s(\mathbf{x}, y)|\mathbf{x}, \theta^*)$.

By Assumption 1, for any $\varepsilon, \delta > 0$, if the calibration set is sufficiently large, then with probability at least $1 - \delta$ (for some event Ω), we have

$$\sup_{s, \mathbf{x}} |F(s \mid \mathbf{x}, D) - F(s \mid \mathbf{x}, \theta^*)| \leq \varepsilon,$$

where the randomness is over D.

Let $\hat{\mathbb{P}}$ denote the empirical probability measure based on $\mathcal{D}_{\mathrm{cal},2}$, i.e., for a given event A and any function g:

$$\hat{\mathbb{P}}(g(s) \in A) = \frac{1}{|\mathcal{D}_{\mathrm{cal},2}|} \sum_{(\mathbf{X}_i, Y_i) \in \mathcal{D}_{\mathrm{cal},2}} \mathbf{1}(g(s(\mathbf{X}_i, Y_i)) \in A).$$

Conditionally on the event Ω, we obtain:

$$1 - \alpha \leq \hat{\mathbb{P}}(F(s|\mathbf{x}, D) \leq t'_{1-\alpha})$$
$$\leq \hat{\mathbb{P}}(F(s|\mathbf{x}, \theta^*) - \varepsilon \leq t'_{1-\alpha}),$$

which, by the definition of the empirical quantile, implies that

$$t''_{1-\alpha} \leq t'_{1-\alpha} + \varepsilon.$$

By a symmetric argument, we also have:

$$1 - \alpha \leq \hat{\mathbb{P}}(F(s|\mathbf{x}, \theta^*) \leq t''_{1-\alpha})$$
$$\leq \hat{\mathbb{P}}(F(s|\mathbf{x}, D) - \varepsilon \leq t''_{1-\alpha}).$$

Therefore, conditionally to the event Ω, of probability at least $1 - \delta$, we conclude that

$$|t'_{1-\alpha} - t''_{1-\alpha}| \leq \varepsilon.$$

Using this result and the fact that $|F(s \mid \mathbf{x}, D) - F(s \mid \mathbf{x}, \theta^*)| \leq \varepsilon$, we establish the following bound:

$$\mathbb{P}(s'(\mathbf{X}, Y) \leq t'_{1-\alpha}|\mathbf{X}) \leq \mathbb{P}(s'(\mathbf{X}, Y) \leq t'_{1-\alpha} \mid \mathbf{X}, \Omega) \cdot 1 + \delta$$
$$\leq \mathbb{P}(s'(\mathbf{X}, Y) \leq t''_{1-\alpha} + \varepsilon \mid \mathbf{X}, \Omega) + \delta$$
$$\leq \mathbb{P}(s''(\mathbf{X}, Y) - \varepsilon \leq t''_{1-\alpha} + \varepsilon|\mathbf{X}, \Omega) + \delta$$
$$= \mathbb{P}(s''(\mathbf{X}, Y) \leq t''_{1-\alpha} + 2\varepsilon \mid \mathbf{X}) + \delta$$
$$= t''_{1-\alpha} + 2\varepsilon + \delta.$$

In the fourth equality, we used the fact that $s''(x, y)$ is non-random and that $t''_{1-\alpha}$ depends only on $\mathcal{D}_{\mathrm{cal},2}$, making it independent of Ω. In the last equality, we used the fact that the random variable $s''(\mathbf{X}, Y)|\mathbf{X}$ is uniform.

By a similar argument, we can show that:

$$\left|\mathbb{P}(s'(\mathbf{X}, Y) \leq t'_{1-\alpha}|\mathbf{X}) - t''_{1-\alpha}\right| \leq 2\varepsilon + \delta.$$

Thus, as $|\mathcal{D}_{\mathrm{cal},1}| \to \infty$, we can take $\varepsilon, \delta \to 0$, which implies that

$$\lim_{|\mathcal{D}_{\mathrm{cal},1}| \to \infty} \mathbb{P}(s'(\mathbf{X}, Y) \leq t'_{1-\alpha}|\mathbf{X}) = t''_{1-\alpha}.$$

Finally, by [Dheur et al., 2025, Lemma 2, Section D.2.2], we know that as $|\mathcal{D}_{\mathrm{cal},2}| \to \infty$, we have $t''_{1-\alpha} \to 1 - \alpha$, which concludes the proof.

$\square$