Conformal Prediction via Regression-as-Classification

Etash Guha
RIKEN Center for AI Project
Tokyo, Japan
{etashguha}@gmail.com

Shlok Natarajan
Salesforce
San Francisco, California, USA
{shloknatarajan}@salesforce.com

Thomas Möllenhoff
RIKEN Center for AI Project
Tokyo, Japan
{thomas.moellenhoff}@riken.jp

Mohammad Emtiyaz Khan
RIKEN Center for AI Project
Tokyo, Japan
{emtiyaz.khan}@riken.jp

Eugene Ndiaye
Apple
Paris, France
{e_ndiaye}@apple.com

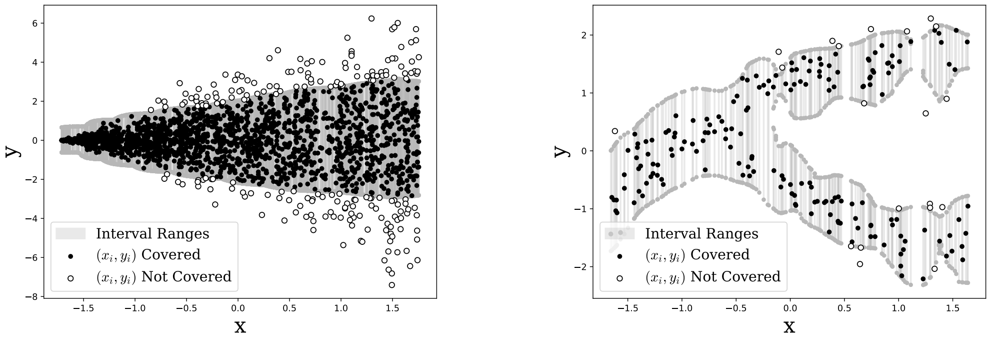
Abstract

Conformal Prediction (CP) is a method of estimating risk or uncertainty when using Machine Learning to help abide by common Risk Management regulations often seen in fields like healthcare and finance. CP for regression can be challenging, especially when the output distribution is heteroscedastic, multimodal, or skewed. Some of the issues can be addressed by estimating a distribution over the output, but in reality, such approaches can be sensitive to estimation error and yield unstable intervals. Here, we circumvent the challenges by converting regression to a classification problem and then use CP for classification to obtain CP sets for regression. To preserve the ordering of the continuous-output space, we design a new loss function and present necessary modifications to the CP classification techniques. Empirical results on many benchmarks shows that this simple approach gives surprisingly good results on many practical problems.

1 Introduction

Quantifying and estimating the uncertainty of machine-learning models is an important task for many problems, especially mission-critical ones requiring trustworthy and reliable predictions. Regulations surrounding the uncertainty or risk of using Machine Learning are prevalent in fields such as healthcare and finance [Black, 2005]. Conformal Prediction (CP) [Vovk et al., 2005] has recently gained popularity and has been used successfully in applications such as breast cancer detection [Lambrou et al., 2009], stroke risk prediction [Lambrou et al., 2010], and drug discovery [Cortés-Ciriano and Bender, 2020]. Under mild conditions, CP techniques aim to construct a prediction set that, for given test inputs, is guaranteed to contain the true (unknown) output with high probability. The set is built using a *conformity score*, which, roughly speaking, indicates the similarity between a new test example and the training examples. The conformal set merely gathers examples that have large conformity scores. Despite its popularity, CP for regression can be challenging, especially when the output distribution is heteroscedastic, multimodal, or skewed [Lei and Wasserman, 2014].

Workshop on Regulatable Machine Learning at the 37th Conference on Neural Information Processing Systems (RegML @ NeurIPS 2023).



(a) Heteroskedastic

(b) Bimodal

Figure 1: We show two examples where the output distribution is heteroskedastic (left) and bimodal (right). In both cases, our method is able to change the interval (shaded gray region) adaptively as the input values x are increased. Examples outside the gray regions (white dots) are deemed different from those inside it (black dots).

The main challenge lies in the design of the *conformity score*. It is common to use a simple choice for score functions such as distance to mean regressor, but such choices may ignore the subtle features of the shape of the output distribution. For instance, this could lead to symmetric intervals or ignoring the heteroscedasticity. In theory, it is better to estimate the (conditional) distribution over the output, for example, by using kernel density estimation and directly using it to build a confidence interval. However, such estimation approaches are also challenging, and estimates can be sensitive to the choice of kernel and hyperparameters, which can yield unstable results. Angelopoulos et al. [2020] We circumvent the challenges by exploiting the existing CP techniques for classification. We proceed by first converting regression to a classification problem and then using CP techniques for classification to obtain a conformal set. Regression-as-classification approaches are popular for various applications in computer vision and have led to more accurate training than only-regression training [Stewart et al., 2023, Zhang et al., 2016, Fu et al., 2018, Rothe et al., 2015, Van Den Oord et al., 2016, Diaz and Marathe, 2019]. We leverage them to construct a distribution-based conformal set that can flexibly capture the shape of the output distribution while preserving the simplicity and efficiency of CP for classification. First, we discretize the output space into bins, treating each bin as a distinct class. Second, to preserve the ordering of the continuous output space, we design an alternative loss function that penalizes the density on bins far from the true output value but also facilitates variability by using an entropy regularization. The loss design is similar in spirit to Weigend and Srivastava [1995], Diaz and Marathe [2019]. The resulting method can adapt to heteroscedasticity, bimodality, or both in the label distribution. We verify this on synthetic and real datasets where we achieve the shortest intervals compared to other CP baselines. See examples in Figure 1.

2 Background on Conformal Prediction

Given a new input x_{new}, CP techniques aim to construct a set that contains the true but unknown output y_{new} with high probability. Assuming that a pair of input-output variables (x, y) has a joint density $p(x, y)$ and a conditional density $p(y \mid x)$, oracle prediction sets (with joint and conditional coverage) for the output y can be constructed as

$$\{y \in \mathbb{R} : p(x, y) \geq \tau_\alpha\} \text{ or } \{y \in \mathbb{R} : p(y \mid x) \geq \tau_{\alpha,x}\}, \tag{1}$$

where the thresholds τ_α and $\tau_{\alpha,x}$ are selected to ensure that the corresponding sets have a probability mass that meets or exceeds prescribed confidence level $1 - \alpha \in (0, 1)$. As the ground-truth distribution is unknown, we rely solely on estimating these uncertainty sets using the density estimators $\hat{p}(x, y)$ and $\hat{p}(y \mid x)$. The latter can be inaccurate due to numerous sources of errors such as model misspecification, small sample size, high optimization errors during training, and overfitting. Without a stronger distribution assumption, the finite-sample guarantee is typically not upheld.

Conformal Prediction has arisen as a method for yielding sets that do hold finite-sample guarantees. Given a partially observed instance $(x_{\text{new}}, y_{\text{new}})$ where y_{new} is unknown, Conformal Prediction (CP) [Vovk et al., 2005] constructs a set of values that contains y_{new} with high probability without knowing the underlying data distribution. Under conformal prediction, this property is guaranteed under the mild assumption that the data satisfies exchangeability. The set is called the conformal set and is built using a conformity score, denoted by $\sigma(x, y)$, which measures how appropriate an output value is for a given input example. There are many ways to build the conformity score, but they all involve splitting the data into a training set $\mathcal{D}_{\text{tr}}$ and a calibration set $\mathcal{D}_{\text{cal}}$. Often, a prediction model $\mu_{\text{tr}}(x)$ is built using the training set, and then a conformity score is obtained using this model along with the calibration set (we will shortly give an example).
The conformal set merely gathers the points with larger conformity scores:

$$\{z \in \mathbb{R} : \sigma(x_{\text{new}}, z) \geq Q_{1-\alpha}(\mathcal{D}_{\text{cal}})\},$$

where $Q_{1-\alpha}(\mathcal{D}_{\text{cal}})$ is the $(1-\alpha)$-quantile of the conformity scores on the calibration data. This set provably contains y_{new} with probability larger than $1-\alpha$ for any finite sample size and without assumption on the ground-truth distribution.

There are many design choices for this conformity score. For example, one can choose a prediction model $\mu_{\text{tr}}(x)$ as an estimate of the conditional expectation and measure conformity as the absolute value of the residual, i.e., $\sigma(x, y) = -|y - \mu_{\text{tr}}(x)|$. The corresponding conformal set is a single interval centered around the prediction $\mu_{\text{tr}}(x)$ and of constant length $Q_{1-\alpha}(\mathcal{D}_{\text{cal}})$ for any example x_{new}, without taking into account its variability. However, in situations where the underlying data distribution demonstrates skewness or heteroscedasticity, we may desire a more flexible conformity score. In order to obtain more adaptive sets, [Lei and Wasserman, 2014] suggested estimating the ground truth distribution using kernel density estimation. In this way, we can capture the shape of the (conditional) distribution while enjoying guaranteed coverage by simply using them as conformity score functions. This, however, not only requires a good choice of kernel but also a good bandwidth parameter selection that controls the regularity of the estimated density. Moreover, [Fong and Holmes, 2021] learns a Bayesian model to learn this density but requires a well-specified prior to work well in practice. Moreover, [Sesia and Romano, 2021] learn many quantile regressors and form a histogram from the quantile regressors to estimate this density. However, their method is susceptible to noise due to the formulation of the density estimate, which needs post hoc linear programming to generate the interval to account for the lack of smoothness. Moreover, [Chernozhukov et al., 2021a] learn a cumulative distributive function through regression techniques. However, their method cannot account for bimodality since it can only output a single interval by design. We wish to generate an approach to approximate this density function that is versatile in the distributions it can represent and avoids the difficulties seen in prior methods. We turn to look at existing tools for density estimation in the Classification Conformal Prediction that already work well.

Related Works Since their introduction, a lot of work has been done to improve the set of conformal predictions. As simple score function, distance to conditional mean ie $\sigma(x, y) = |y - \mu(x)|$ where $\mu(x)$ is an estimate of $\mathbb{E}(y \mid x)$ was prominently used [Papadopoulos et al., 2002, Lei et al., 2018, Guha et al., 2023]. Instead, [Romano et al., 2019] suggests estimating a conditional quantile instead and a conformity score function based on the distance from a trained quantile regressor, i.e. $\sigma(x, y) = \max(\mu_{\alpha/2}(x) - y, y - \mu_{1-\alpha/2}(x))$ where μ_α are the α-th quantile regressors. Within the literature, our strategy is more closely related to the distribution-based methods [Lei et al., 2013]. Following a similar line of work [Chernozhukov et al., 2021b] argued that the conformal quantile regression score function might be less adaptive since the distance of the quantile behaves similarly to the distance of the mean estimate. Instead, they suggested estimating the cumulative (conditional) distribution function and directly outputting $\{y : F(x, y) \in [\alpha/2, 1-\alpha/2]\}$. An equally interesting approach for distribution-based prediction sets is based on learning a Bayesian estimator, which, however, may require a well-specified prior and can be computationally expensive [Fong and Holmes, 2021]. Variants based on estimating the conditional density of the response using histogram regressors [Sesia and Romano, 2021] could detect a possible asymmetry of the ground-truth distribution. However, our densities are estimated through classification techniques, whereas their densities are learned through a histogram of many regressors. Furthermore, smoothness over the distribution is encoded in our loss function, whereas they use a post hoc subinterval finding algorithm through linear programming to prevent many disjoint intervals.

3 CP via Regression-as-Classification

3.1 Classification Conformal Prediction

We aim to compute a conformity function that accurately predicts the appropriateness of a label for a specific data point. Given that the distribution function across labels can adopt diverse forms, such as being bimodal, heavy-tailed, or heteroscedastic, our approach must effectively factor in such shapes while upholding its coverage precision. A frequently employed technique involves using the conditional label density as a conformity function, leading to reliable results in the classification context. Typically, practitioners perform conformal prediction for classification with probability estimates from a Softmax neural network that covers K output logits using cross-entropy loss.

Let us denote the parametrized density

$$q_\theta(\cdot \mid x) = \mathrm{softmax}(f_\theta(x)), \text{ where } \mathrm{softmax}(v)_j = \frac{\exp(v_j)}{\sum_{k=1}^K \exp(v_k),}$$

as the outputted discrete probability distribution over the labels of the input x.

Traditionally, we can train our neural network by minimizing the cross-entropy loss on the training dataset, i.e., solving $\hat{\theta} = \arg\min_\theta \sum_{i=1}^n \mathrm{KL}(\delta_{y_i} \mid\mid q_\theta(\cdot \mid x_i))$. Let us assume that we have trained and acquired a $\hat{\theta}$ that has minimized the traditional cross-entropy loss function on the training dataset. A natural conformity score is simply the probability of a label according to the learned conditional distribution, i.e., $\sigma(x, y) = q_{\hat{\theta}}(y \mid x)$. This approach is both straightforward and efficient. The neural network's flexibility allows it to learn numerous label distributions with adaptivity across examples without explicitly designing specific prior information or structure.

3.2 Regression to Classification Approach

Naturally, we strive to embody such practicality and effectiveness in regression settings. However, the distribution of labels in the regression scenario is continuous, and learning a continuous distribution directly using a neural network is challenging [Rothfuss et al., 2019]. Often, Bayesian or kernel density estimators are employed to estimate this distribution. Other techniques acquire knowledge of this distribution by training numerous regressors and categorizing the regressors, for example, Conformal Histogram Regression [Sesia and Romano, 2021].

However, these methods look different from the conformal prediction approaches for classification. It would be desirable to be able to use similar methods for both classification and regression conformal prediction. One method of unifying classification and regression problems outside the conformal prediction literature is known as *Regression-as-Classification*. We simply turn a regression problem into a classification problem by binning the range space. Specifically, we generate K bins with K equally spaced numbers covering the interval $\mathcal{Y} = [y_{\min}, y_{\max}]$, where $y_{\min}$ (or $y_{\max}$) is the minimum (or maximum) value of the labels observed in the training set. More explicitly, we define our discretization of the label space as

$$\hat{\mathcal{Y}} = \{\hat{y}_1, \ldots, \hat{y}_K\} \text{ where } \hat{y}_{k+1} = \hat{y}_k + \frac{\hat{y}_K - \hat{y}_1}{K-1} \text{ with } \hat{y}_1 = y_{\min} \text{ and } \hat{y}_K = y_{\max}.$$

These values $\hat{y} \in \hat{\mathcal{Y}}$ form the midpoints for each bin of our discretization. Naturally, the kth bin is all the labels in the range space $\mathcal{Y}$ closest to $\hat{y}_k$. Intuitively, we can treat each bin as a class. Thus, we have turned a regression problem into a classification problem. This method is simple but has yielded surprising results. Some work has suggested that this form of binning results in more stable training [Stewart et al., 2023] and gives significantly better results for learning conditional expectations. To unify classification and regression conformal prediction, a simple solution is to employ the Classification Conformal Prediction model with discrete labels $\tilde{y}_i = \arg\min_{\hat{y} \in \hat{\mathcal{Y}}} |y_i - \hat{y}|$. This will aid in training the neural network with modified labels through cross-entropy loss, resulting in a discrete distribution of $q_\theta(\cdot \mid x)$, as outlined in the previous section.

To compute conformity scores for all labels, we employ linear interpolation from the discrete probability function $q_\theta(\cdot \mid x)$ to generate the continuous distribution $\bar{q}_\theta(\cdot \mid x)$. Nevertheless, this approach encounters a critical issue when employed for regression.

3.3 Data Fitting

A critical problem with employing CrossEntropy loss in the classification conformal prediction context is that *any structural relationships between classes are disregarded.* This is not surprising given that in the classification context, no structure exists between classes, and each class is independent. Therefore, the CrossEntropy loss does not need to differentiate between whether q_θ allocates probable mass far away or close to the actual label. Instead, it only incentivizes the allocation of probable mass on the correct label. However, within the regression setting, despite a multitude of labels, the labels adhere to an ordinal structure. Thus, to enhance the accuracy of labeling, it is imperative to devise a loss function that incentivizes the allocation of probabilistic mass not only to the correct bin but also to the neighboring bins. Formally, given an input and output pair (x, y), our goal is to determine a density estimate $q_\theta(\cdot \mid x)$ that assigns low (resp. high) probability to points that are far (resp. close) to the true label y, i.e.,

$$q_\theta(\hat{y} \mid x_i) \text{ high (resp. small) when the loss } \ell(\hat{y}, y_i) \text{ is small (resp. high).}$$

Hence, a natural desideratum for learning the probability density function q_θ is that their product $\ell(y, \hat{y})q(\hat{y} \mid x)$ is small in expectation. We propose to find a distribution q_θ minimizing the loss

$$\mathbb{E}_{\hat{y} \sim q(\cdot|x)}[\ell(y, \hat{y})] = \sum_{k=1}^{K} \ell(y, \hat{y}_k) q(\hat{y}_k \mid x)$$

Minimizing this loss in the space of all possible distributions $q(\cdot \mid x)$ is equivalent to minimizing the original loss $\ell(y, \cdot)$, where the minimizing distribution is a Dirac delta $\delta_{\hat{y}_\star}$ at the minimizer $\hat{y}_\star$ of $\ell(y, \cdot)$. Therefore, we expect an unregularized version of this loss to share similarities with the typical empirical risk minimization on $\ell(y, \cdot)$. However, a key difference is that when minimizing in a restricted family of distributions (for example, those representable by a neural network with a fixed architecture), the distributional output can represent multi-modal or heavy-tailed label distributions. Minimizing the original loss $\ell(y, \cdot)$, one would always be confined to a point estimate.

3.4 Entropic Regularization

Although the proposed loss function better encodes the connection between bins, it tends towards outputting Dirac distributions, as outlined in the previous paragraph. Overconfidence in our neural network is a commonly reported problem in the literature [Wei et al., 2022]. Nevertheless, smoothness has been a traditional requirement in density learning. As such, we rely on a classical entropy-regularization technique for learning density estimators [Wainwright and Jordan, 2008]. Given the set of density estimators that match the training label distribution well and put high probability mass on the best bins, we prefer the probability distribution that maximizes the entropy since this intuitively takes fewer assumptions on the data distribution structure. Our choice is based on selecting density estimators that effectively match the distribution of the training labels and assign a higher probability to the best bins. Formally, we can calculate the entropy of our probability distribution by using the Shannon entropy $\mathcal{H}$ of the produced probability distribution $q(\cdot \mid x)$ as a penalty term as follows:

$$\mathcal{H}(q_\theta(\cdot \mid x)) = \sum_{k=1}^{K} q_\theta(\hat{y}_k|x) \log q_\theta(\hat{y}_k|x).$$

Summary We learn a distribution by minimizing the following expected loss over a given training data $\mathcal{D}_{\mathrm{tr}} = \{(x_1, y_1), \ldots, (x_{n_{\mathrm{tr}}}, y_{n_{\mathrm{tr}}})\}$ of size n_{tr}:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n_{\mathrm{tr}}} \sum_{k=1}^{K} \ell(y_i, \hat{y}_k)\, q_\theta(\hat{y}_k \mid x_i) - \tau \mathcal{H}(q_\theta(\cdot \mid x_i)), \tag{2}$$

In particular, we choose $\ell(y_i, \hat{y}_k)$ as $\ell(y_i, \hat{y}_k) = |y_i - \hat{y}_k|^p$ where $p > 0$ is a hyperparameter. This selection functions as a natural distance metric that meets all required objectives. As a result, we can employ a technique similar to the aforementioned Classification Conformal Prediction methods. Initially, we train a Softmax neural network f_θ with K logits, grounded on the loss function Equation (2) on the training dataset. To calculate conformity scores for the calibration set, we utilize the linearly interpolated $\sigma(x, y) = \bar{q}_\theta(y \mid x)$ for a specific data point (x, y).
Complete details of this procedure are outlined in Algorithm 1.

Algorithm 1 Regression to Classication Conformal Prediction (R2CCP).

```
Input:
    • Dataset $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and new input $x_{n+1}$
    • Desired confidence level $\alpha \in (0, 1)$
Hyperparameters: temperature $\tau > 0$, $p > 0$, number of bins $K > 1$
Discretize the output space $[y_{\text{min}}, y_{\text{max}}]$ into $K$ equidistant bins with midpoints $\{\hat{y}_1, \ldots, \hat{y}_K\}$
Randomly split the dataset $\mathcal{D}_n$ in training $\mathcal{D}_{\text{tr}}$ and calibration $\mathcal{D}_{\text{cal}}$
Find a distribution $q_{\hat{\theta}}(\cdot \mid x)$ by (approximately) optimizing on the training set $\mathcal{D}_{\text{tr}}$
    $\hat{\theta} \in \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n_{\text{tr}}} \sum_{k=1}^{K} |y_i - \hat{y}_k|^p q_\theta(\hat{y}_k \mid x_i) - \tau \mathcal{H}(q_\theta(\cdot \mid x_i))$
    where $q_\theta(\hat{y}_k \mid x) = \text{softmax}(f_\theta(x))_k$ for a model (e.g., neural net) $f_\theta : \mathbb{R}^d \to \mathbb{R}^K$.
$\mathcal{S} \leftarrow \left\{\bar{q}_{\hat{\theta}}(y \mid x) \text{ for } (x, y) \in \mathcal{D}_{\text{cal}}\right\}$ # $\bar{q}_{\hat{\theta}}(\cdot \mid x)$ is linear interpolation of softmax probabilities.
$Q_\alpha(\mathcal{D}_{\text{cal}}) \leftarrow \text{quantile}(\mathcal{S}, \alpha)$
return Conformal Set $\Gamma^{(\alpha)}(x_{n+1}) = \{z \in \mathbb{R} \mid \bar{q}_{\hat{\theta}}(z \mid x_{n+1}) \geq Q_\alpha(\mathcal{D}_{\text{cal}})\}$
```

4 Experiments

We investigate the empirical behavior of our `R2CCP` (Regression-to-Classification Conformal Prediction) method, which we have explained in detail in Algorithm 1. We have three sets of experiments. The first one is described in Section 4.1 and presents empirical evidence of the algorithm's ability to produce narrower intervals by utilizing label density characteristics, including heteroscedasticity, bimodality, or a combination. Section 4.2 demonstrates the effectiveness of our algorithm on synthetic and real data by comparing it with various benchmarks from the Conformal Prediction literature in terms of length and coverage. Section 4.3 evaluates the effect of different loss functions on the final learned densities and their impact on the intervals produced.

4.1 Specific Characteristics of Label Distribution

Heteroscedascity We generate a toy dataset where the input is one-dimensional. It contains samples from the following distribution: $y \sim \mathcal{N}(0, |x|)$. The label distribution is heteroscedastic, meaning the variance of the labels changes as the input changes. In traditional Conformal Prediction literature, many existing algorithms fail to capture heteroscedasticity, resulting in wide intervals for inputs x where the label distribution of y has low variance. However, our learned algorithm can directly learn this relation and adjust the outputted probability distribution accordingly. Thus, we see that the lengths of the intervals will increase as the variance of the label distribution increases, which is desirable. We see this relation in Figure 1a. Moreover, we also use the dataset generated from [Lei and Wasserman, 2014] as discussed in Appendix A. This dataset exhibits heteroskedascity and bimodality as X passes -0.5. We see that our learned method can adjust intervals accordingly to maintain coverage and length for all X. We plot this in Figure 1b. We plot how the intervals (grey) change as the data distribution (black) changes. As the variance of the labels increases as x increases, the produced intervals adaptively get wider, taking advantage of the heteroscedasticity.

Bimodality We showcase our algorithm's capability to address labels with a bimodal distribution. Our bimodal dataset is generated by repeatedly (1) sampling two sets of random features that are close geometrically and (2) giving one set of features a label of 1 plus some Gaussian noise and giving the other a label of -1 plus some Gaussian noise. Therefore, our dataset is comprised of many similar data points with bimodally distributed labels. This bimodal distribution is particularly hard for many existing CP algorithms to solve since it requires outputting two disjoint conformal sets to achieve low length. CQR, for instance, cannot deal with this circumstance and will generate a conformal set that covers the entire range space. However, our method is flexible enough to assign a low probability to labels between -1 and 1, and our resulting conformal set will not include these intermediate labels. We see this in Figure 3d. In Figure 3d, our outputted probability distribution has two modes around labels -1 and 1 and assigns a low probability value to the valley between the modes.

Dataset	Bimodal	LogNorm	Concrete	MEPS-19	MEPS-20	MEPS-21	Bio	Community
CQR	$2.14_{(0.01)}$	$1.58_{(0.03)}$	$\mathbf{0.39_{(0.01)}}$	$1.87_{(0.05)}$	$2.00_{(0.07)}$	$1.99_{(0.03)}$	$1.34_{(0.01)}$	$\mathbf{1.44_{(0.03)}}$
KDE	$\mathbf{0.35_{(0.01)}}$	$\mathbf{1.40_{(0.04)}}$	$1.54_{(0.03)}$	$2.16_{(0.02)}$	$2.51_{(0.05)}$	$2.39_{(0.03)}$	$2.27_{(0.00)}$	$2.23_{(0.09)}$
Lasso	$2.14_{(0.00)}$	$3.30_{(0.06)}$	$2.74_{(0.03)}$	$4.64_{(0.06)}$	$4.79_{(0.04)}$	$4.92_{(0.04)}$	$3.89_{(0.00)}$	$3.26_{(0.05)}$
CB	$2.16_{(0.00)}$	$1.45_{(0.01)}$	$0.96_{(0.01)}$	$4.47_{(0.02)}$	$4.50_{(0.02)}$	$4.51_{(0.01)}$	$2.09_{(0.00)}$	$1.80_{(0.00)}$
CHR	$2.14_{(0.00)}$	$1.52_{(0.05)}$	$0.47_{(0.02)}$	$2.60_{(0.08)}$	$2.53_{(0.03)}$	$2.75_{(0.03)}$	$1.59_{(0.01)}$	$\mathbf{1.49_{(0.05)}}$
R2CCP (Ours)	$0.46_{(0.01)}$	$1.96_{(0.03)}$	$\mathbf{0.38_{(0.01)}}$	$\mathbf{1.60_{(0.01)}}$	$\mathbf{1.70_{(0.03)}}$	$\mathbf{1.72_{(0.03)}}$	$\mathbf{1.11_{(0.01)}}$	$\mathbf{1.47_{(0.03)}}$

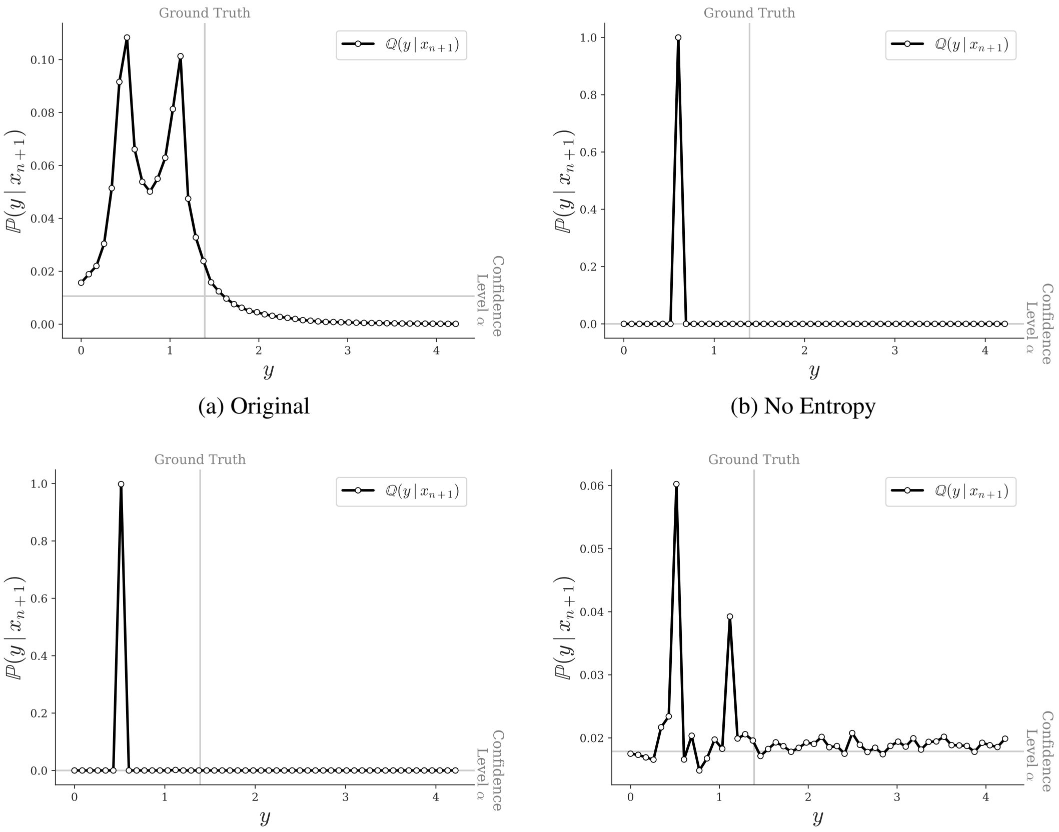
Dataset	Diabetes	Solar	Parkinsons	Stock	Cancer	Pendulum	Energy	Forest
CQR	$1.30_{(0.05)}$	$1.98_{(0.22)}$	$\mathbf{0.42_{(0.01)}}$	$1.85_{(0.23)}$	$\mathbf{3.09_{(0.13)}}$	$2.25_{(0.31)}$	$\mathbf{0.19_{(0.01)}}$	$3.18_{(0.19)}$
KDE	$1.34_{(0.05)}$	$\mathbf{0.50_{(0.01)}}$	$3.79_{(0.02)}$	$4.72_{(0.29)}$	$3.82_{(0.09)}$	$3.96_{(0.09)}$	$2.72_{(0.07)}$	$\mathbf{2.90_{(0.17)}}$
Lasso	$3.01_{(0.04)}$	$3.54_{(0.12)}$	$3.46_{(0.03)}$	$1.39_{(0.04)}$	$3.55_{(0.14)}$	$3.99_{(0.07)}$	$1.29_{(0.03)}$	$3.97_{(0.29)}$
CB	$\mathbf{1.19_{(0.01)}}$	$3.78_{(0.02)}$	$3.42_{(0.01)}$	$1.32_{(0.01)}$	$\mathbf{3.14_{(0.04)}}$	$3.71_{(0.03)}$	$1.26_{(0.01)}$	$3.75_{(0.03)}$
CHR	$1.40_{(0.02)}$	$1.49_{(0.23)}$	$0.68_{(0.02)}$	$1.59_{(0.07)}$	$3.42_{(0.12)}$	$\mathbf{1.69_{(0.11)}}$	$0.23_{(0.01)}$	$\mathbf{3.03_{(0.15)}}$
R2CCP (Ours)	$1.34_{(0.02)}$	$3.80_{(2.61)}$	$0.50_{(0.00)}$	$\mathbf{0.92_{(0.02)}}$	$\mathbf{3.21_{(0.08)}}$	$\mathbf{1.60_{(0.07)}}$	$\mathbf{0.20_{(0.02)}}$	$3.80_{(0.26)}$

Table 1: This is the length results over all datasets. We see that our method achieves the best length on 10 of the 16 datasets. Meanwhile, CQR is best at 5, CHR is best at 3, CB is best at 1, and KDE is the best at 3. Our method achieves the shortest intervals across these datasets.

4.2 Comparison to other Conformal Prediction Algorithms

The crucial criteria for assessing a Conformal Prediction algorithm consist of (1) coverage, representing the percentage of generated conformal sets that incorporate accurate labels, and (2) the length of the generated conformal sets. Our baseline techniques consist of the Kernel Density Estimator as proposed by Lei and Wasserman [2014], alongside a conformity score shown by the estimated probability density. Furthermore, we take into consideration Fong and Holmes [2021], which employs the likelihood of a posterior distribution in their conformity function, and Sesia and Romano [2021], which uses quantile regressors on every bin of a histogram density estimator. We have included the Conformal Quantile Regression as described by Romano et al. [2019], which employs conformity based on the labels' distance from quantile regressors. Additionally, we have used the Lasso Conformal Predictor with a distance to mean regressors, which is the most straightforward option. We use synthetic data exhibiting bimodally distributed and log-normally distributed to illustrate particular weaknesses of existing methods. We use real datasets also in Romano et al. [2019]. Specifically, these are several datasets from the UCI Machine Learning repository (Bio, Blog, Concrete, Community, Energy, Forest, Stock, Cancer, Solar, Parkinsons, Pendulum) [Nottingham et al., 2023] and the medical expenditure panel survey number 19–21 (MEPS-19–21) [Cohen et al., 2009]. These regression datasets are commonly used to evaluate the strength of regression models. Our approach yields tighter intervals on real datasets than some of the strongest existing baselines.

Results We report lengths and coverages results in Table 1 respectively. We added figures depicting example probability distributions on these datasets in Figure 3 in Appendix D. The intervals produced by our method are the shortest over 10 of the 16 datasets. From Figure 3, we see that our method can learn many different shaped distributions well, which accounts for this significant improvement in intervals. Overall, the Kernel Density Estimation and our method can accurately predict the best intervals on datasets where the connection between the data and feature is simple, such as Bimodal or Log-Normal. While the Kernel Density Estimator can fail when the labels are complexly related to inputs, no such connection exists in these datasets since the labels are independent of the features. Thus, the Kernel Density Method's simplicity allows it to learn the label density quickly. Our method also seems to handle the case where there is no connection between the data and the labels, as seen in Figure 3d in Appendix D. Moreover, on datasets where the label density is smooth and close to the Laplace prior, such as on Figure 3c and Figure 3i, Conformal Bayes and our method can accurately learn this distribution since both methods can output smooth distributions. Moreover, on very sharp datasets such as on Concrete in Figure 3j and Energy in Figure 3h, both our method and CQR can capture the sharp and unnoisy distribution needed to achieve strong length. Moreover, on complex distributions such as in Pendulum in Figure 3e and Bio in Figure 3f, we see that both CHR and our method have the flexibility to portray complex distributions resulting in the most accurate intervals. Thus, the flexibility of our algorithm to smoothly learn sharp, wide, complex, and simple conditional label densities results in our method achieving the best length most consistently over the entire dataset.



(a) Original

(b) No Entropy

(c) MLE

(d) MLE with Entropy

Figure 2: The resulting density estimates with different loss functions. We see that removing entropy from our loss function or using MLE as error terms causes sharp density estimates. Moreover, adding in the entropy regularization with MLE does not smooth the density estimate but instead raises the entire distribution uniformly. This does not give any valuable information for the Conformal Prediction process.

4.3 Ablation Studies

Our loss function consists of an error and entropy terms. The error term penalizes distributions that put weight far away from the true label, whereas the entropy term acts as a regularizing term in the probability distribution space. We will do ablations on both the error and entropy terms to illustrate the importance of each part. For the error term, there are several notable alternatives that a practitioner may use. An alternative is the log maximum likelihood or cross-entropy formulations of the error term, which we denote as

$$\mathcal{L}_{\text{MLE}}(\theta) = -\sum_{i=1}^{N} \log(q_\theta(\tilde{y}_i \mid x_i)),$$

where we remind the reader that $\tilde{y}_i = \arg\min_{\hat{y} \in \hat{\mathcal{Y}}} |\hat{y} - y_i|$. We note that both MLE and CE are equivalent formulations in this setting. These two are standard error terms often used in practice. We will train our models with this loss function over all datasets to see how the change of error term affects the intervals' length and the learned density. We will show that our chosen error term is better for producing optimal intervals empirically. Moreover, we will also demonstrate the importance of our entropy term. We do this by retraining our models with the entropy part omitted. We also test by combining the MLE error term with entropy regularization. Therefore, we will perform 4 different ablation experiments on the loss function by retraining on different variations of the loss functions and reporting the final length and coverage generated. Overall, the four loss functions we will use are the original loss function $\mathcal{L}$, the original loss function without entropy $\mathcal{L}_{\text{NE}}$, the MLE loss function

DATASET	BIMODAL	LOG-NORMAL	CONCRETE	MEPS-19	MEPS-20	MEPS-21	BIO	COMMUNITY
$\mathcal{L}_{\text{NE}}$	$0.43_{0.001}$	$3.25_{0.058}$	$1.89_{0.037}$	$3.73_{0.057}$	$3.79_{0.061}$	$8.78_{0.163}$	$2.01_{0.006}$	$3.64_{0.054}$
$\mathcal{L}_{\text{MLE}}$	$\mathbf{0.35_{0.001}}$	$\mathbf{1.76_{0.020}}$	$0.74_{0.016}$	$\mathbf{1.58_{0.008}}$	$\mathbf{1.59_{0.009}}$	$1.71_{0.034}$	$2.71_{0.001}$	$2.32_{0.047}$
$\mathcal{L}_{\text{MLE + E}}$	$0.36_{0.001}$	$3.52_{0.008}$	$1.91_{0.017}$	$\mathbf{1.60_{0.008}}$	$1.63_{0.027}$	$\mathbf{1.70_{0.008}}$	$2.32_{0.003}$	$3.77_{0.013}$
$\mathcal{L}$	$0.44_{0.002}$	$1.82_{0.034}$	$\mathbf{0.37_{0.004}}$	$\mathbf{1.60_{0.009}}$	$\mathbf{1.59_{0.011}}$	$\mathbf{1.69_{0.013}}$	$\mathbf{1.10_{0.004}}$	$\mathbf{1.50_{0.025}}$

DATASET	DIABETES	SOLAR	PARKINSONS	STOCK	CANCER	PENDULUM	ENERGY	FOREST
$\mathcal{L}_{\text{NE}}$	$1.92_{0.052}$	$2.20_{0.233}$	$4.76_{0.033}$	$10.17_{0.165}$	$3.26_{0.207}$	$12.82_{0.479}$	$3.30_{0.093}$	$3.33_{0.254}$
$\mathcal{L}_{\text{MLE}}$	$1.56_{0.031}$	$\mathbf{0.14_{0.005}}$	$0.34_{0.006}$	$4.48_{0.214}$	$3.26_{0.117}$	$3.00_{0.203}$	$0.25_{0.011}$	$3.04_{0.110}$
$\mathcal{L}_{\text{MLE + E}}$	$1.96_{0.012}$	$0.15_{0.005}$	$\mathbf{0.23_{0.001}}$	$9.73_{0.074}$	$3.95_{0.022}$	$13.40_{0.221}$	$0.25_{0.009}$	$5.24_{0.055}$
$\mathcal{L}$	$\mathbf{1.37_{0.037}}$	$0.66_{0.092}$	$0.46_{0.036}$	$\mathbf{1.95_{0.039}}$	$\mathbf{3.04_{0.142}}$	$\mathbf{1.62_{0.042}}$	$\mathbf{0.21_{0.025}}$	$\mathbf{2.92_{0.127}}$

Table 2: we present the length results over all of the variant loss functions. We find that our loss function delivers the best over 12 datasets, demonstrating that our chosen loss function often generates the best intervals. For datasets where our method does not deliver the best results, it is likely that tuning the weight on the entropy τ and the smoothing term p would likely have improved the results, but we do not do this for the sake of evaluation.

$\mathcal{L}_{\text{MLE}}$, and the MLE loss function with entropy added $\mathcal{L}_{\text{MLE + E}}$. We will demonstrate that our chosen loss function delivers the best results across all the loss functions. We report our results in Table 2.

Results We see that our chosen loss function achieves the best length across most loss functions in Table 2. The only datasets where our chosen loss function does not achieve the best length are Energy, Solar, Bimodal, and Log-Normal. We now discuss the individual differences between our loss function and each variant loss function. When comparing our loss $\mathcal{L}$ with the variant without entropy $\mathcal{L}_{\text{NE}}$, we see the lengths constantly increase except for the bimodal distribution. Without an entropy regularizing term, the outputted probability distributions are not smooth, placing much mass on a single data point. This overconfidence results in the αth quantile of the low probability of the true label as in Figure 2. When looking at the differences between our loss function when changing the error term to Maximum Log Likelihood, we see similar overconfidence. While MLE loss works on several datasets, such as Energy, Solar, Bimodal, and Log-Normal, it also performs poorly on other datasets. MLE works well for datasets with less simple label distributions but fails otherwise due to similar overconfidence. Even when adding entropy as a regularizer to the MLE loss as in $\mathcal{L}_{\text{MLE + E}}$, we see that the addition of entropy does not improve the length. Since MLE loss does not treat nearby bins differently than far away bins, regularization decreases the overconfidence but does so uniformly across all bins. The result is a roughly uniform distribution with a single spike. This does not utilize the structure of regression where bins near the best bin are preferable. Thus, prioritizing nearby bins is crucial for achieving strong length. Moreover, when adding entropy regularization to our error term, we do not see a uniform increase in probability across all bins but instead a smoothening of a direct. This connection between entropy regularization and our distance-based loss function appears to be a powerful synergy per the intervals produced. Thus, the error term utilizing the regression structure by prioritizing nearby bins and the regularizing entropy term preventing overconfidence seem crucial for producing good intervals.

Another important insight is that the datasets where other loss functions excel are the same ones where other CP methods perform better than ours. In particular, on the Bimodal, Log-Normal, Parkinson's, and Solar datasets, our CP method is not superior, and other loss functions outperform it. This indicates that the MLE's sharp distributions result in greater accuracy on these specific datasets. Hence, adjusting the weighting terms for entropy τ and smoothing p could enhance the smoothness of the learned distribution and, thereby, the performance of our algorithm across these datasets, leading to the best outcomes. Nevertheless, to maintain fairness, we avoid this approach.

5 Limitations

There are several limitations to our algorithm. The loss function we choose to optimize needs larger representational power due to increased complexity of the output. Therefore, the neural networks we used in practice to minimize this loss function are larger than the ones used for CQR. Moreover, to achieve strong training, we had to slightly vary the hyperparameters for a dataset depending on its size. For example, larger datasets needed models to be trained with smaller initial learning rate to avoid divergence. This was not needed for Quantie Regressors. If the network does not train well or achieve good training or validation loss, the intervals produced will be suboptimal, so finding a training setup that minimizes the loss effectively is crucial for our algorithm to produce meaningful intervals. Moreover, we do not directly learn the label distribution and our produced probability distributions do not directly mirror that of the ground truth distribution. One reason for this is the use of entropy regularization. While using entropy regularization introduces bias, we found it to be necessary to prevent the neural network from being overconfident on a simple bin in practice.

References

Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In International Conference on Learning Representations, 2020.

Julia Black. The emergence of risk-based regulation and the new public risk management in the united kingdom. Public law, (3):512–548, 2005.

V. Chernozhukov, K. Wüthrich, and Y. Zhu. An exact and robust conformal inference method for counterfactual and synthetic controls. Journal of the American Statistical Association, 2021a.

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

Joel W Cohen, Steven B Cohen, and Jessica S Banthin. The medical expenditure panel survey: a national information resource to support healthcare cost research and inform policy and practice. Medical Care, pages 44–50, 2009.

Isidro Cortés-Ciriano and Andreas Bender. Concepts and applications of conformal prediction in computational drug discovery. Artificial Intelligence in Drug Discovery, pages 63–101, 2020.

Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

Edwin Fong and Chris C Holmes. Conformal Bayesian computation. In Advances in Neural Information Processing Systems (NeurIPS), 2021.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

Etash Kumar Guha, Eugene Ndiaye, and Xiaoming Huo. Conformalization of sparse generalized linear models. In International Conference on Machine Learning (ICML), 2023. URL `https://proceedings.mlr.press/v202/guha23b.html`.

Antonis Lambrou, Harris Papadopoulos, and Alex Gammerman. Evolutionary conformal prediction for breast cancer diagnosis. In 2009 9th international conference on information technology and applications in biomedicine, pages 1–4. IEEE, 2009.

Antonis Lambrou, Harris Papadopoulos, Efthyvoulos Kyriacou, Constantinos S Pattichis, Marios S Pattichis, Alexander Gammerman, and Andrew Nicolaides. Assessment of stroke risk based on morphological ultrasound image analysis with conformal prediction. In Artificial Intelligence Applications and Innovations: 6th IFIP WG 12.5 International Conference, AIAI 2010, Larnaca, Cyprus, October 6-7, 2010. Proceedings 6, pages 146–153. Springer, 2010.

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, James Robins, and Larry Wasserman. Distribution-free prediction sets. Journal of the American Statistical Association, 108(501):278–287, 2013.

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.

Kolby Nottingham, Rachel Longjohn, and Markelle Kelly. UCI Machine Learning Repository, 2023. URL `https://archive.ics.uci.edu/datasets`. Accessed: September, 2023.

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

Y. Romano, E. Patterson, and E. J. Candes. Conformalized quantile regression. In Advances in Neural Information Processing Systems (NeurIPS), 2019.

Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In IEEE International Conference on Computer Vision Workshops (ICCVW), 2015.

Jonas Rothfuss, Fabio Ferreira, Simon Walther, and Maxim Ulrich. Conditional density estimation with neural networks: Best practices and benchmarks. arXiv preprint arXiv:1903.00954, 2019.

Matteo Sesia and Yaniv Romano. Conformal prediction using conditional histograms. In Advances in Neural Information Processing Systems (NeurIPS), 2021.

Lawrence Stewart, Francis Bach, Quentin Berthet, and Jean-Philippe Vert. Regression as classification: Influence of task formulation on neural network features. In International Conference on Artificial Intelligence and Statistics (AISTATS), 2023.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In International Conference on Machine Learning (ICML), 2016.

V. Vovk, A. Gammerman, and G. Shafer. Algorithmic learning in a random world. Springer, 2005.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning, 1(1–2):1–305, 2008.

Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In International Conference on Machine Learning, pages 23631–23644. PMLR, 2022.

Andreas S Weigend and Ashok N Srivastava. Predicting conditional probability distributions: A connectionist approach. International Journal of Neural Systems, 6(02):109–118, 1995.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In European Conference on Computer Vision (ECCV), 2016.

A Details on Dataset from Lei and Wasserman [2014]

For the dataset referenced in Figure 1b, we generate the dataset by sampling many (X, Y) pairs from the following distribution.

$$\begin{aligned} X &\sim \text{Unif}[-1.5, 1.5] \\ Y \mid X &\sim \frac{1}{2}\mathcal{N}(f(X) - g(X), \sigma^2(X)) + \frac{1}{2}\mathcal{N}(f(X) + g(X), \sigma^2(X)) \\ f(X) &= (X-1)^2(X+1),\ g(X) = 4\sqrt{(X+1/2)\mathbb{I}(X \geq -1/2)},\ \sigma^2(X) = 1/4 + |X| \end{aligned}$$

This dataset demonstrates bimodality after x passes the threshold of -0.5. This dataset was similarly used by the Lei and Wasserman [2014].

B Coverage Data

We have presented the coverage data for the Conformal Prediction comparison as well as the ablation experiments in Table 3 and Table 4 respectively. Since all methods obey the classical Conformal Prediction framework, we expect coverage at the guaranteed level of $1-\alpha$ from **??**. This guaranteed is indeed what we see across all experiments. This confirms that our coverage guarantee indeed holds in practice.

DATASET	BIMODAL	LOGNORM	CONCRETE	MEPS-19	MEPS-20	MEPS-21	BIO	COMMUNITY
CQR	$0.90_{(0.00)}$	$0.91_{(0.00)}$	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$
KDE	$0.93_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.91_{(0.00)}$	$0.92_{(0.00)}$	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$
LASSO	$0.90_{(0.00)}$	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$
CB	$0.90_{(0.00)}$	$0.91_{(0.01)}$	$0.88_{(0.01)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.89_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$
CHR	$0.90_{(0.00)}$	$0.91_{(0.00)}$	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$
R2CCP (OURS)	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$

DATASET	DIABETES	SOLAR	PARKINSONS	STOCK	CANCER	PENDULUM	ENERGY	FOREST
CQR	$0.91_{(0.00)}$	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.91_{(0.00)}$	$0.92_{(0.00)}$	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.91_{(0.00)}$
KDE	$0.91_{(0.00)}$	$0.71_{(0.15)}$	$0.89_{(0.00)}$	$0.90_{(0.00)}$	$0.92_{(0.01)}$	$0.90_{(0.01)}$	$0.91_{(0.00)}$	$0.81_{(0.08)}$
LASSO	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.91_{(0.00)}$	$0.92_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$
CB	$0.89_{(0.00)}$	$0.92_{(0.01)}$	$0.90_{(0.00)}$	$0.88_{(0.01)}$	$0.87_{(0.02)}$	$0.88_{(0.01)}$	$0.89_{(0.01)}$	$0.90_{(0.01)}$
CHR	$0.91_{(0.00)}$	$0.91_{(0.00)}$	$0.90_{(0.00)}$	$0.91_{(0.00)}$	$0.92_{(0.00)}$	$0.91_{(0.00)}$	$0.91_{(0.00)}$	$0.91_{(0.00)}$
R2CCP (OURS)	$0.90_{(0.00)}$	$0.92_{(0.02)}$	$0.96_{(0.00)}$	$0.92_{(0.02)}$	$0.90_{(0.00)}$	$0.90_{(0.00)}$	$0.91_{(0.01)}$	$0.89_{(0.00)}$

Table 3: This is the coverage data over all datasets. We see that all methods achieve the roughly $1-\alpha$ coverage guaranteed by conformal prediction.

DATASET	BIMODAL	LOGNORM	CONCRETE	MEPS-19	MEPS-20	MEPS-21	BIO	COMMUNITY
$\mathcal{L}$	$0.90_{(0.02)}$	$0.90_{(0.02)}$	$0.90_{(0.02)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.90_{(0.00)}$	$0.90_{(0.01)}$
$\mathcal{L}_{\text{NE}}$	$0.90_{(0.02)}$	$0.81_{(0.03)}$	$0.84_{(0.03)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.68_{(0.01)}$	$0.88_{(0.02)}$
$\mathcal{L}_{\text{MLE}}$	$0.90_{(0.02)}$	$0.90_{(0.02)}$	$0.90_{(0.02)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.90_{(0.00)}$	$0.90_{(0.01)}$
$\mathcal{L}_{\text{MLE + E}}$	$0.90_{(0.02)}$	$0.90_{(0.02)}$	$0.90_{(0.02)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.90_{(0.01)}$	$0.92_{(0.00)}$	$0.90_{(0.01)}$

DATASET	DIABETES	SOLAR	PARKINSONS	STOCK	CANCER	PENDULUM	ENERGY	FOREST
$\mathcal{L}$	$0.90_{(0.03)}$	$0.90_{(0.02)}$	$0.95_{(0.01)}$	$0.90_{(0.03)}$	$0.90_{(0.05)}$	$0.90_{(0.03)}$	$0.90_{(0.03)}$	$0.89_{(0.03)}$
$\mathcal{L}_{\text{NE}}$	$0.94_{(0.02)}$	$0.90_{(0.02)}$	$0.73_{(0.01)}$	$0.89_{(0.03)}$	$0.90_{(0.05)}$	$0.90_{(0.03)}$	$0.90_{(0.02)}$	$0.77_{(0.04)}$
$\mathcal{L}_{\text{MLE}}$	$0.90_{(0.03)}$	$0.90_{(0.02)}$	$0.90_{(0.01)}$	$0.91_{(0.03)}$	$0.90_{(0.05)}$	$0.90_{(0.03)}$	$0.90_{(0.02)}$	$0.89_{(0.03)}$
$\mathcal{L}_{\text{MLE + E}}$	$0.90_{(0.03)}$	$0.90_{(0.02)}$	$0.90_{(0.01)}$	$0.90_{(0.03)}$	$0.90_{(0.05)}$	$0.90_{(0.03)}$	$0.90_{(0.03)}$	$0.90_{(0.03)}$

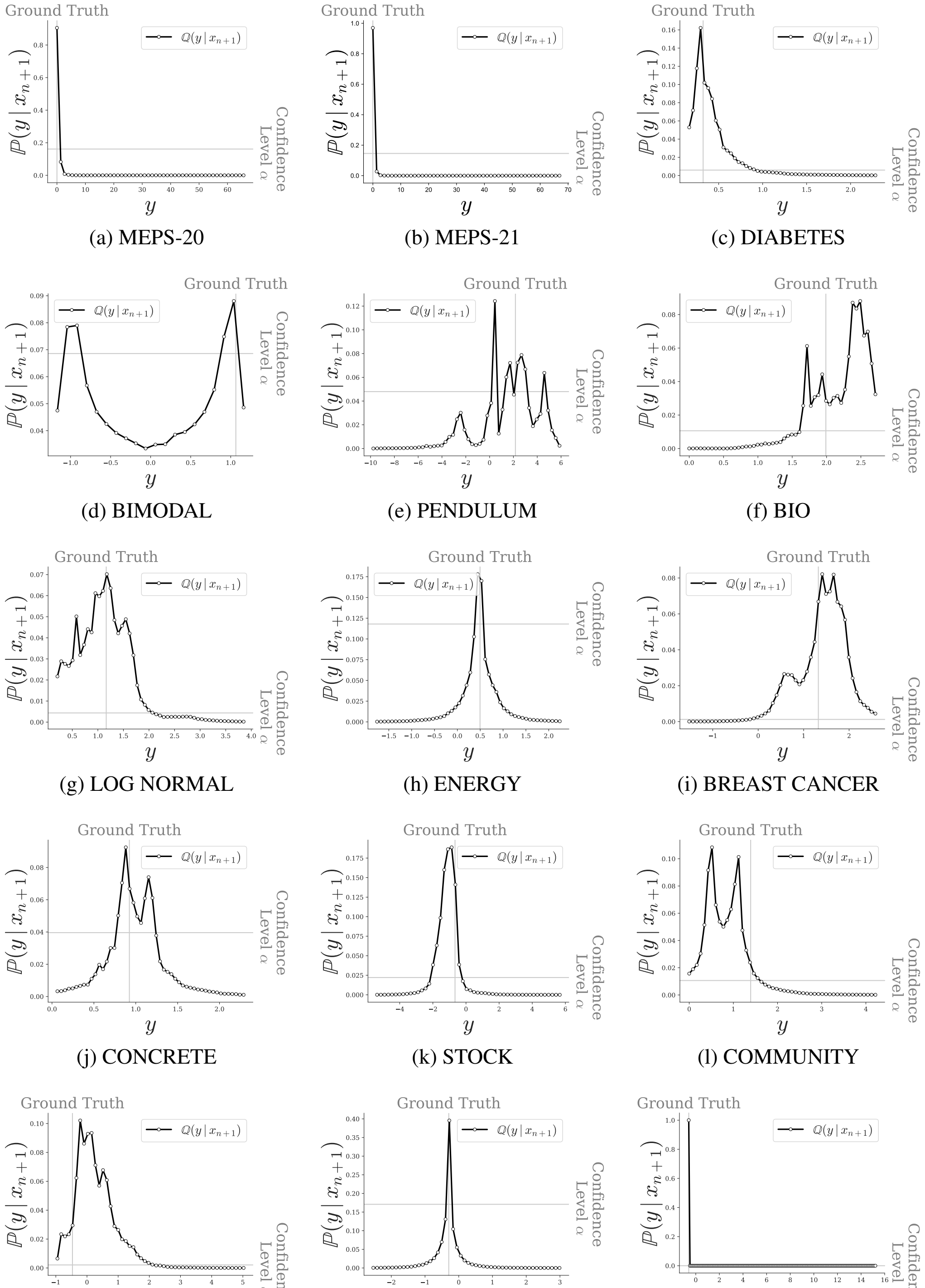
Table 4: This is the coverage data for different loss functions from the ablation.

C More Experimental Details

In order to maintain fairness across all baselines, we use the same size neural network for our method across all experiments. Specifically, we discretize the range space into $K = 50$ points, weight the entropy term by $\tau = 0.2$, use a 1000 hidden dimension, use 4 layers, use weight decay of $1e-4$, use $p = .5$, and use AdamW as an optimizer. For most of the experiments, we use learning rate $1e-4$ and batch size 32. However, for certain datasets, namely the MEPS datasets, we used a larger batch size of 256 to improve training time and used a smaller learning rate to prevent training divergence. We did change any other parameter between all of our runs. For the baselines of CQR and CHR, we use the neural network configurations mentioned in the paper. We found that the parameterizations mentioned by the authors in the papers achieved the best performance and changing the parameterizations weakened their results.

D Output Distribution

Here, we plot the distribution of lengths of all Conformal Prediction methods over all datasets. We also plot example density functions learned by out method, CHR, and KDE on all datasets. We note that KDE's learned densities seem to be relatively less informative. Moreover, the learned densities from CHR are noisy and not smooth. Moreover, we plot several examples of only our learned probability distribution. We use this when referencing the experiments to demonstrate the different shapes of label distributions from the data.



(a) MEPS-20

(b) MEPS-21

(c) DIABETES

(d) BIMODAL

(e) PENDULUM

(f) BIO

(g) LOG NORMAL

(h) ENERGY

(i) BREAST CANCER

(j) CONCRETE

(k) STOCK

(l) COMMUNITY

(m) FOREST

(n) PARKINSON'S

(o) SOLAR

Figure 3: Example Probability Distributions outputted by our learned methodology on different datapoints from different datasets. We can see there are a variety of different shapes of distributions learned.

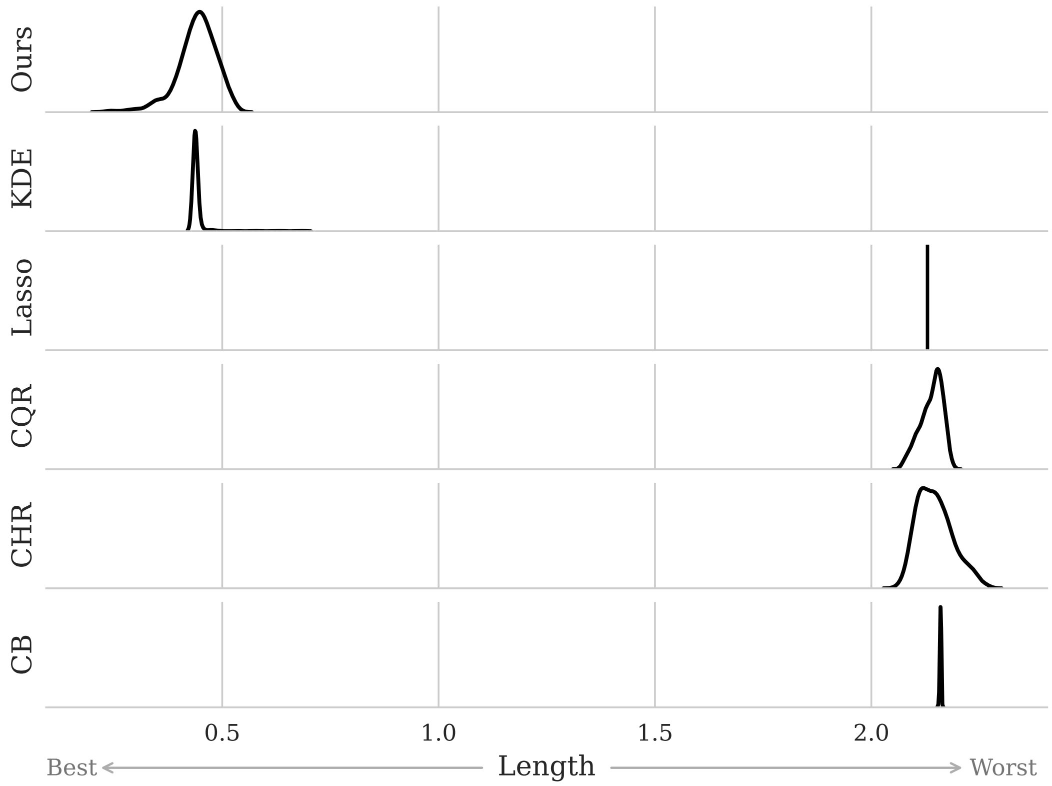


Figure 4: Bimodal Length Distribution across all CP Methods

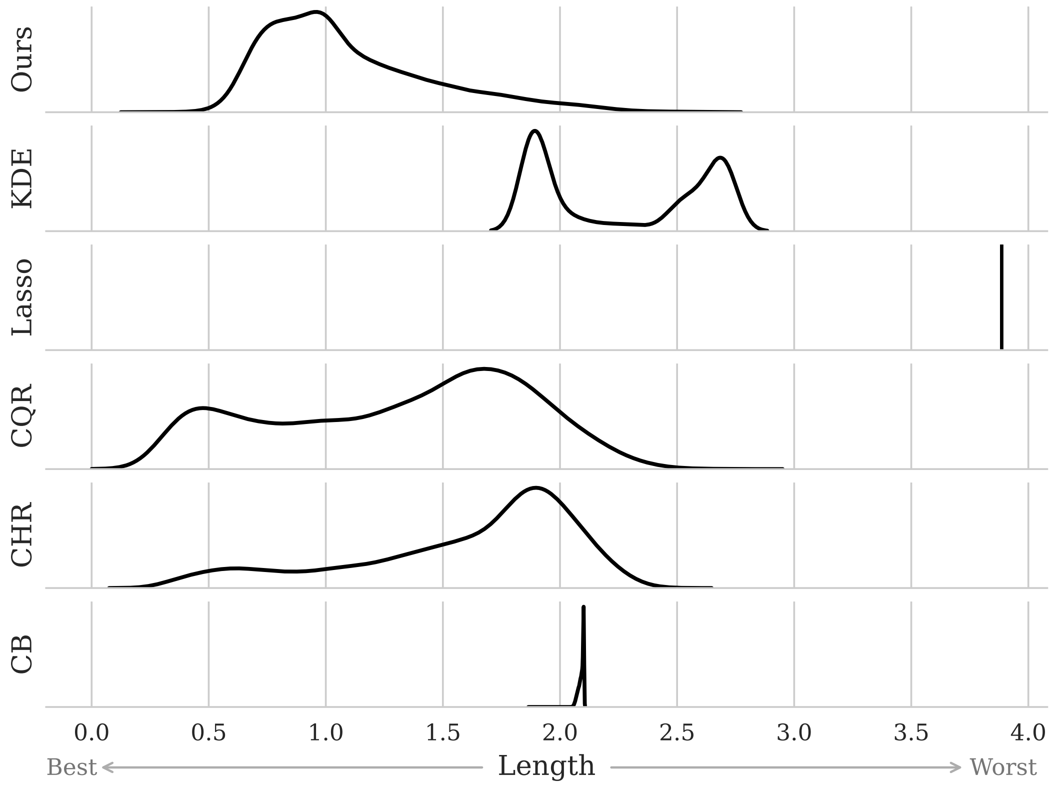


Figure 5: Bio Length Distribution across all CP Methods

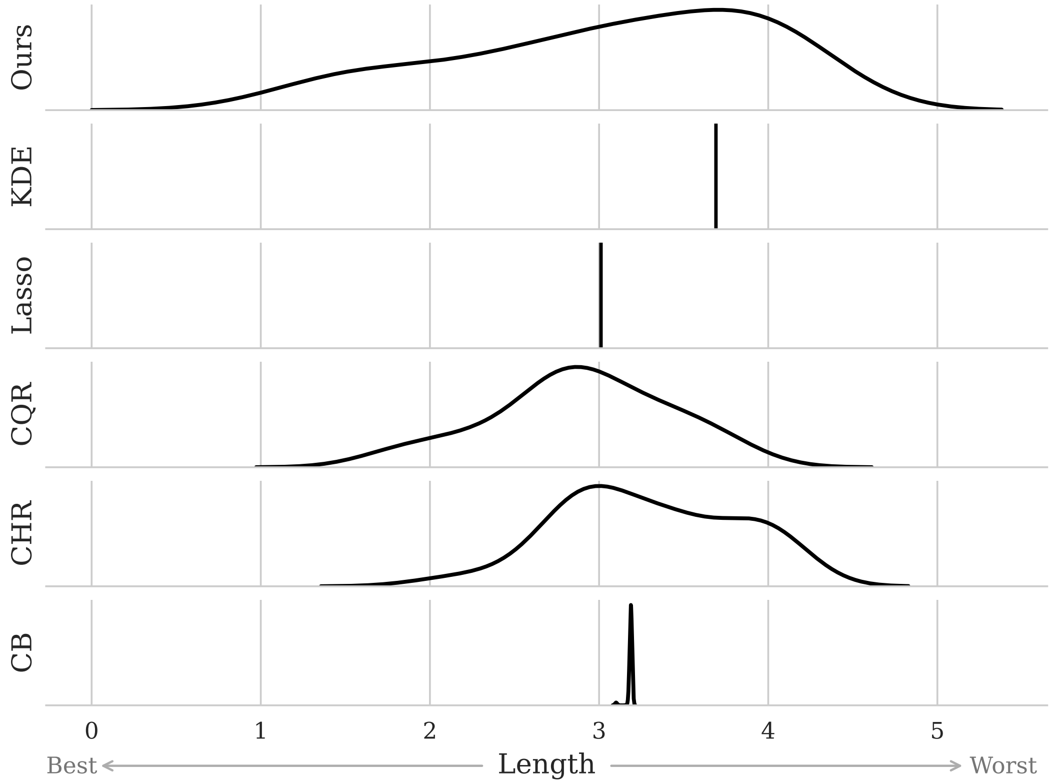


Figure 6: Breast Cancer Length Distribution across all CP Methods

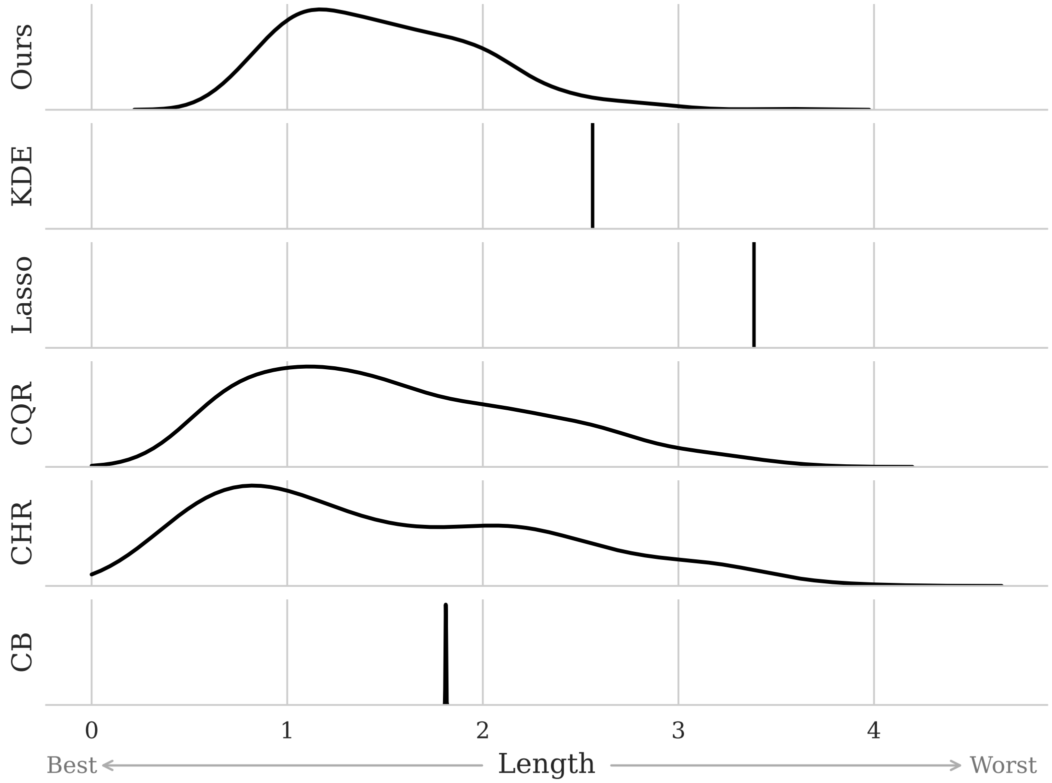


Figure 7: Community Length Distribution across all CP Methods

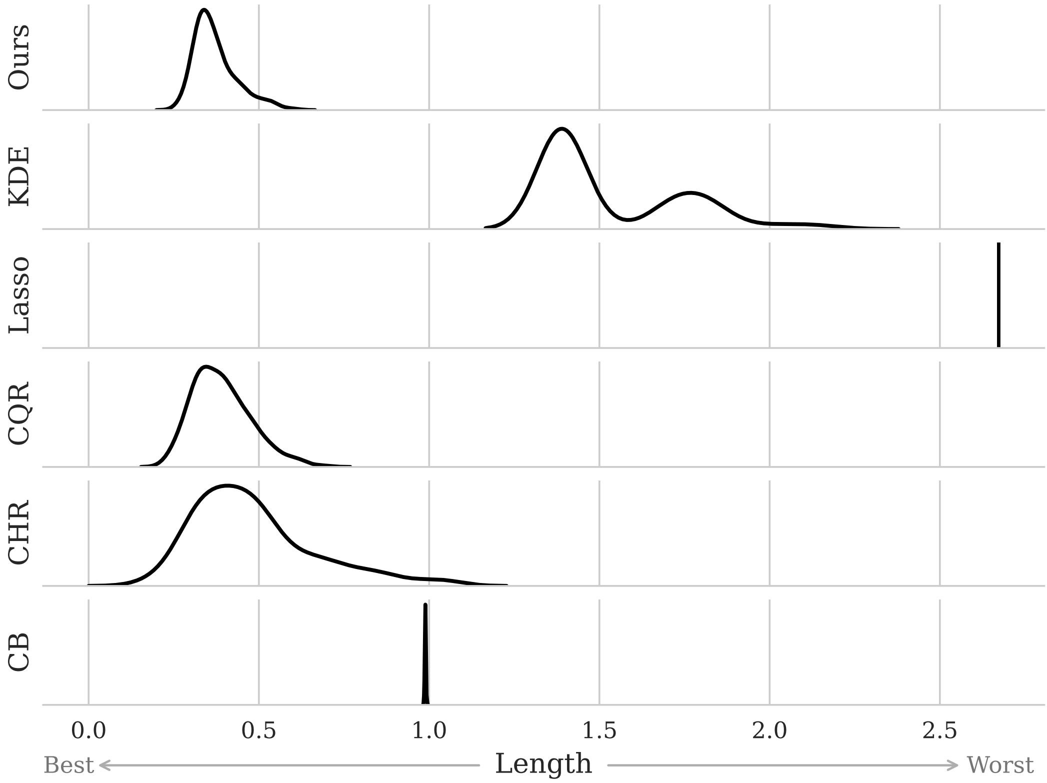


Figure 8: Concrete Length Distribution across all CP Methods

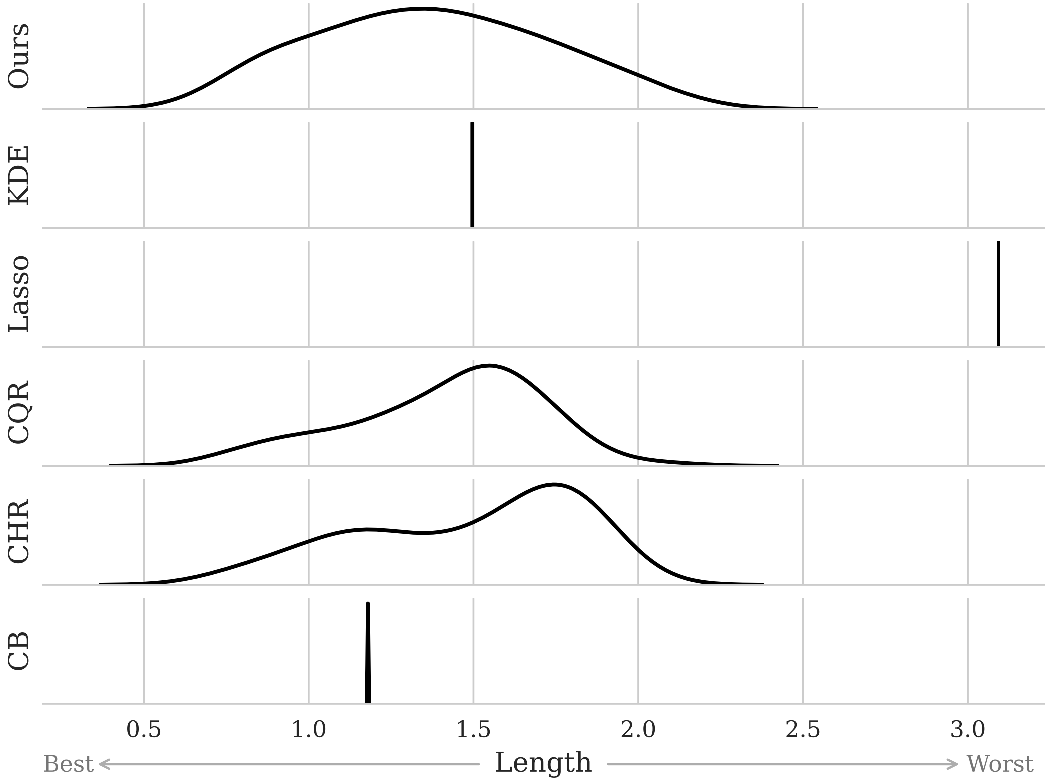


Figure 9: Diabetes Length Distribution across all CP Methods

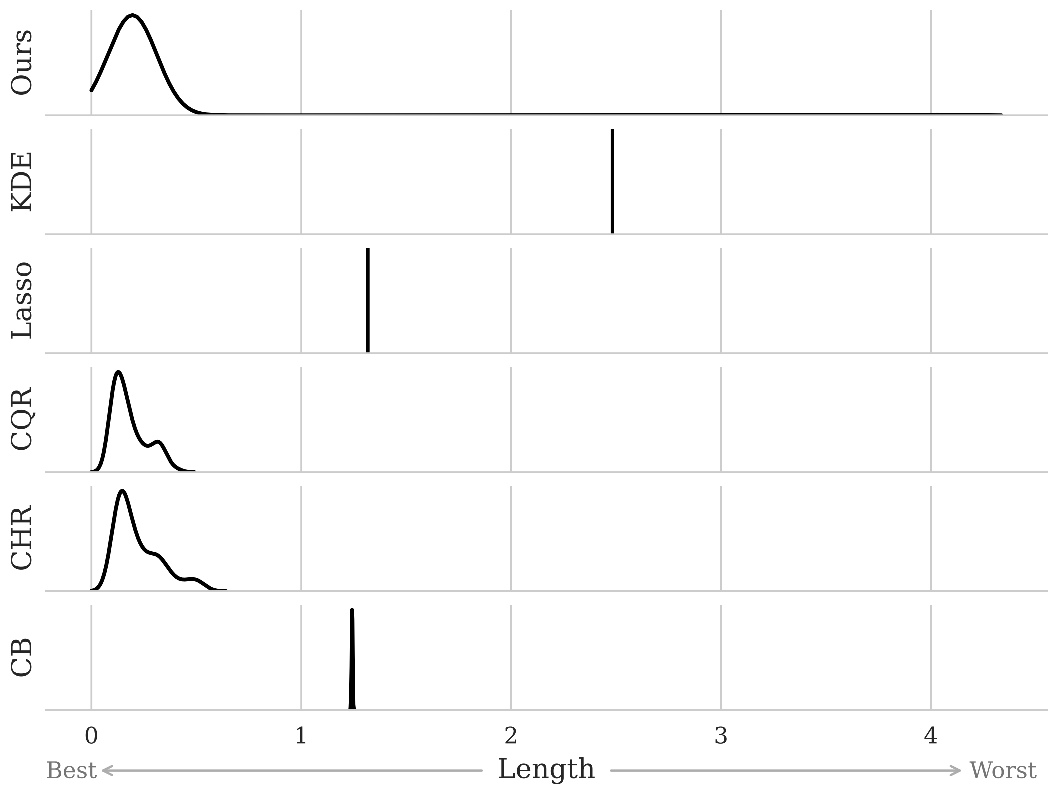


Figure 10: Energy Length Distribution across all CP Methods

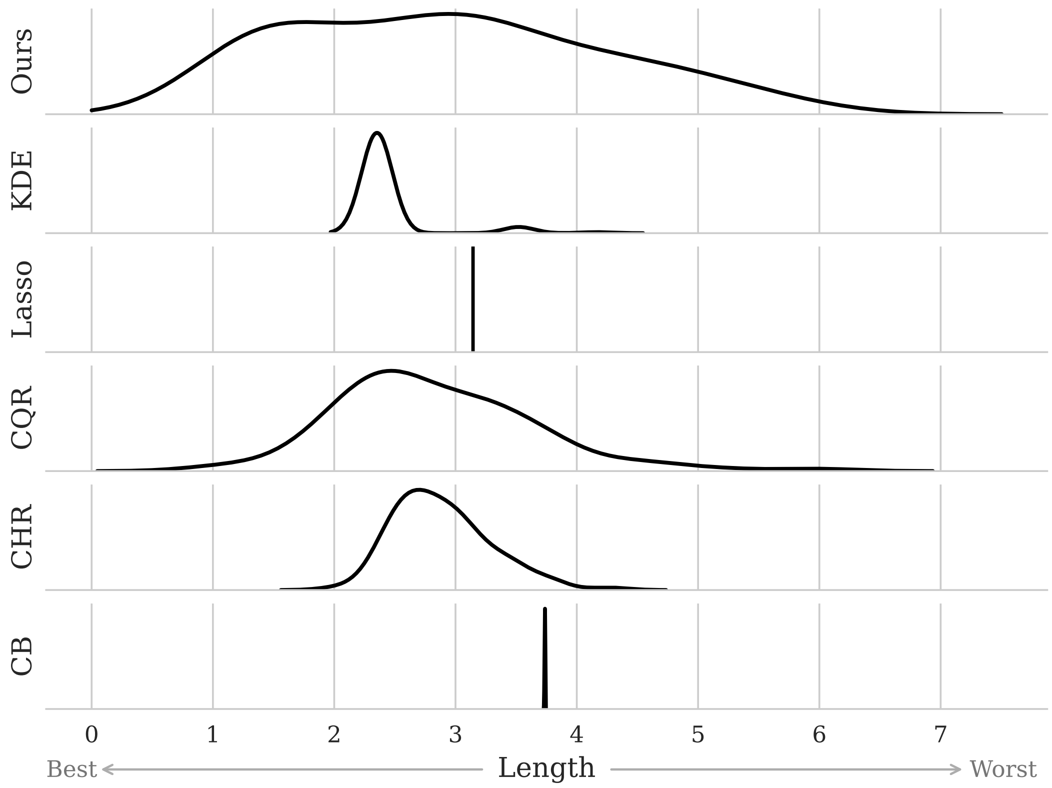


Figure 11: Forest Length Distribution across all CP Methods

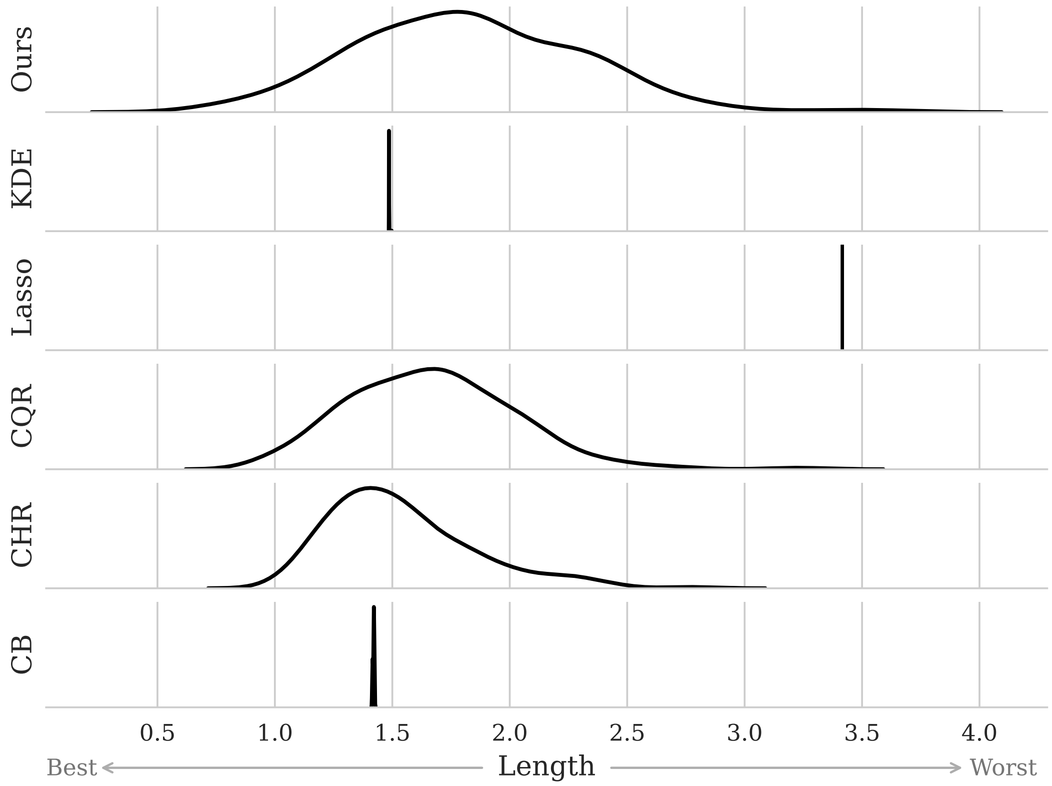


Figure 12: Log Normal Length Distribution across all CP Methods

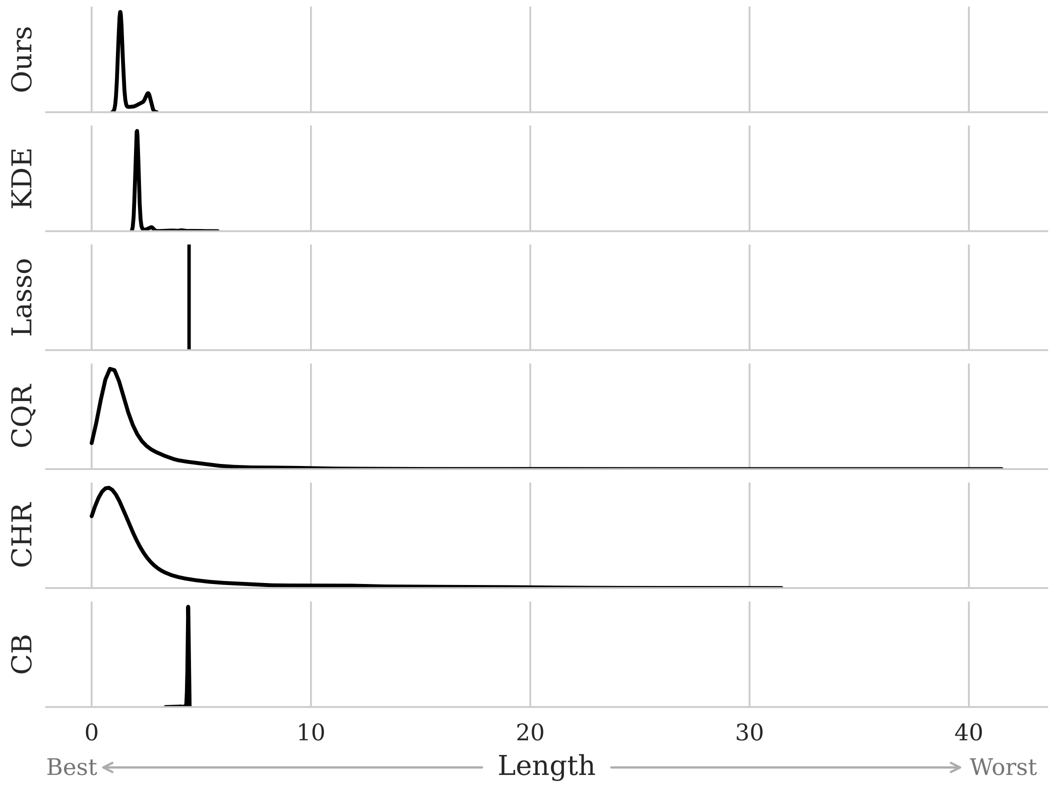


Figure 13: Meps 19 Length Distribution across all CP Methods

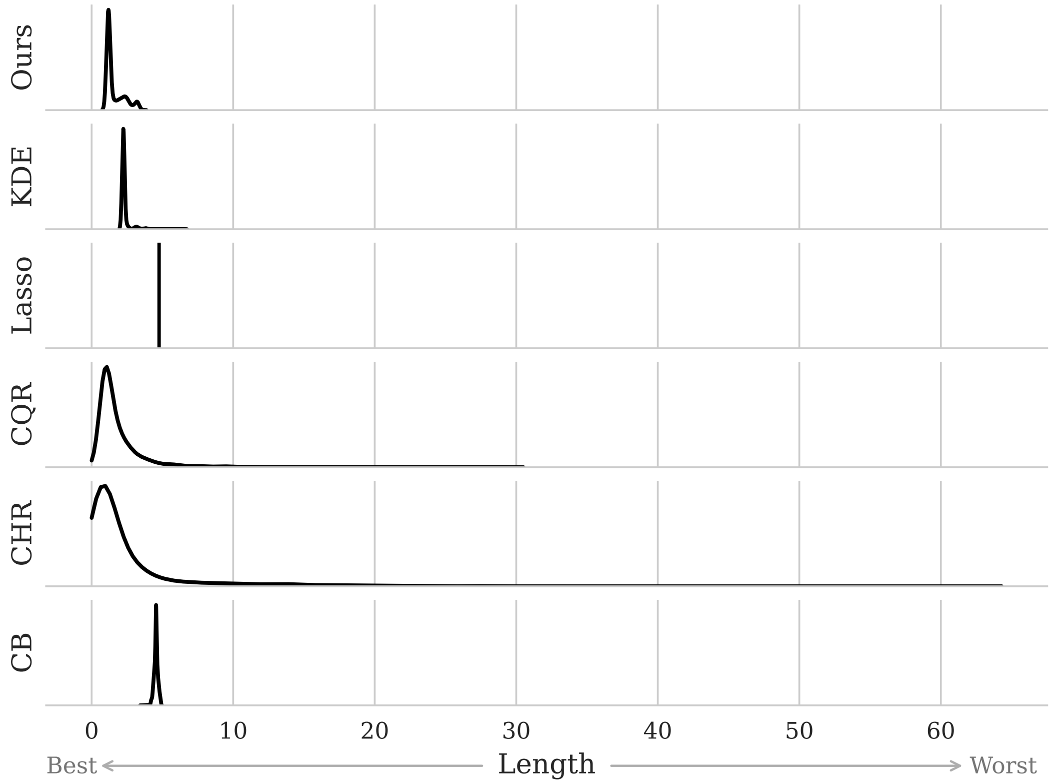


Figure 14: Meps 20 Length Distribution across all CP Methods

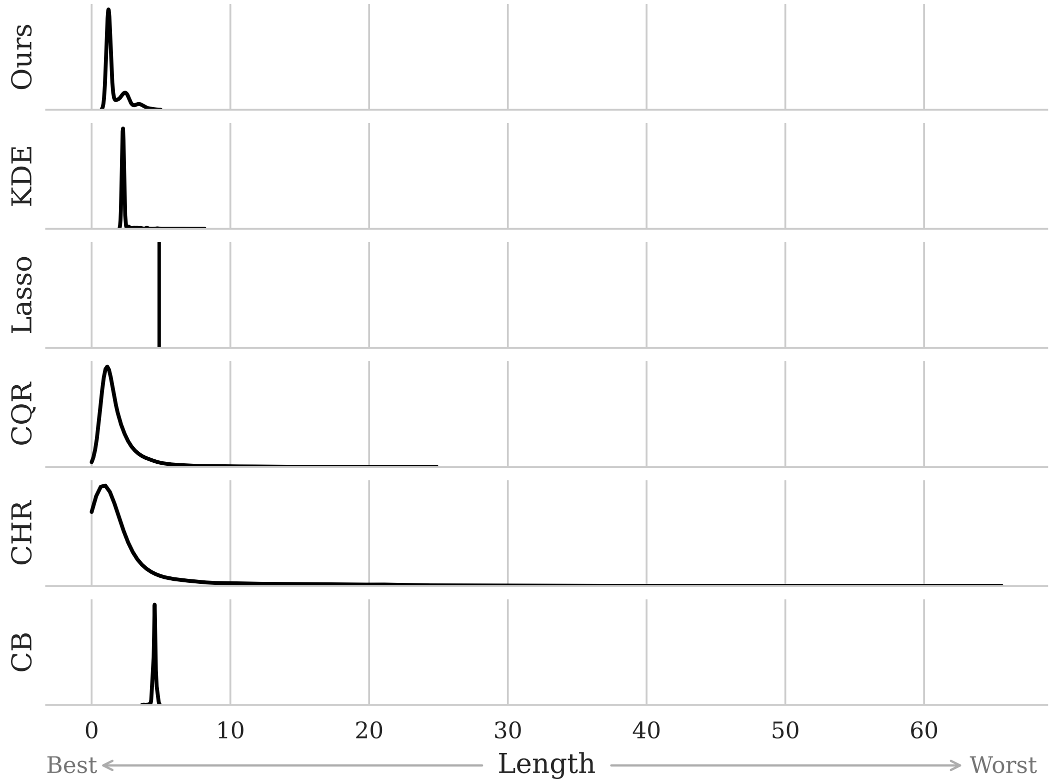


Figure 15: Meps 21 Length Distribution across all CP Methods

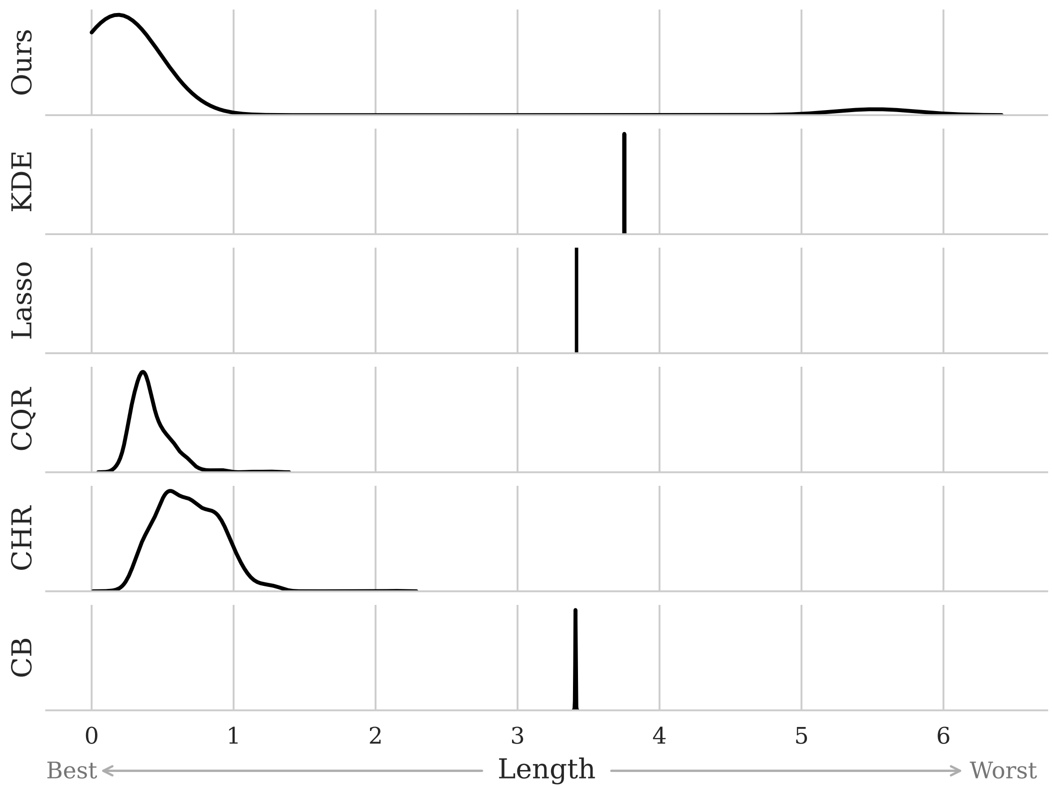


Figure 16: Parkinsons Length Distribution across all CP Methods

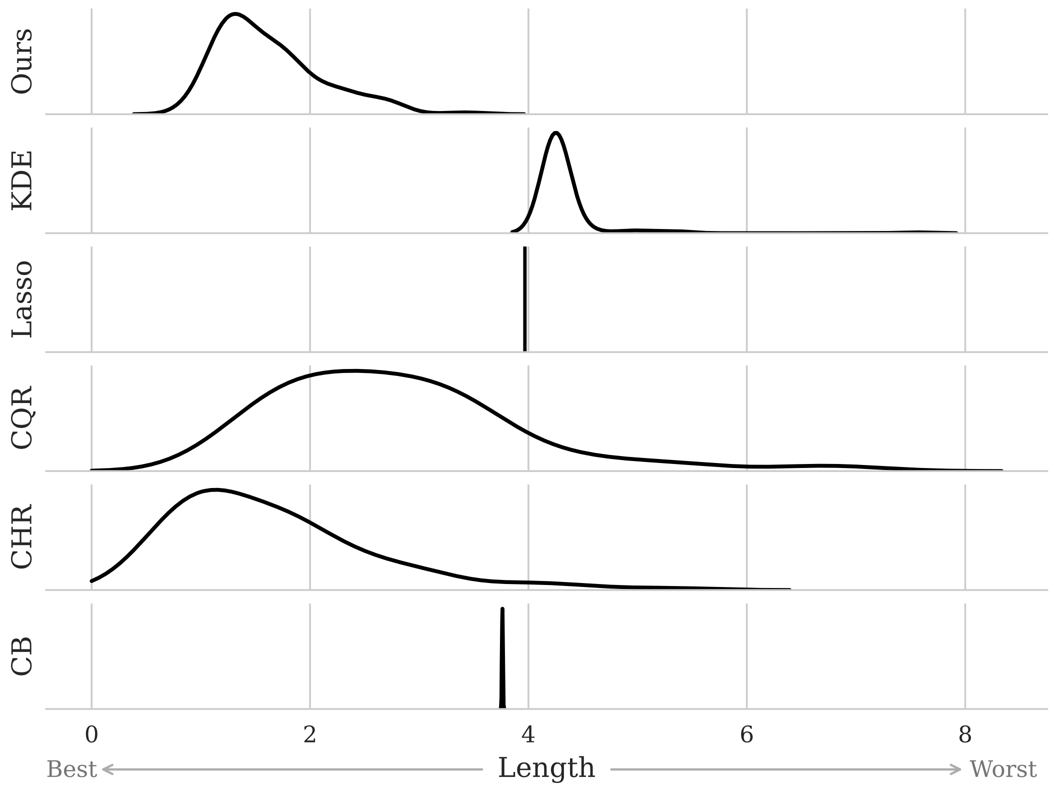


Figure 17: Pendulum Length Distribution across all CP Methods

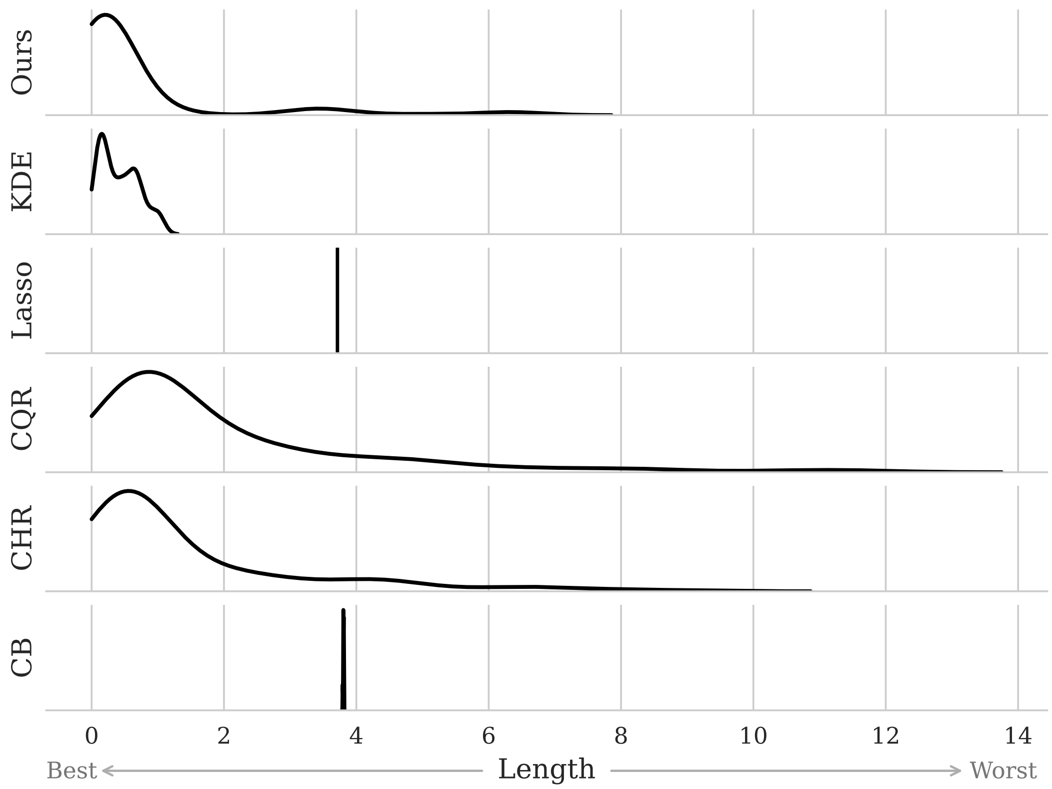


Figure 18: Solar Length Distribution across all CP Methods

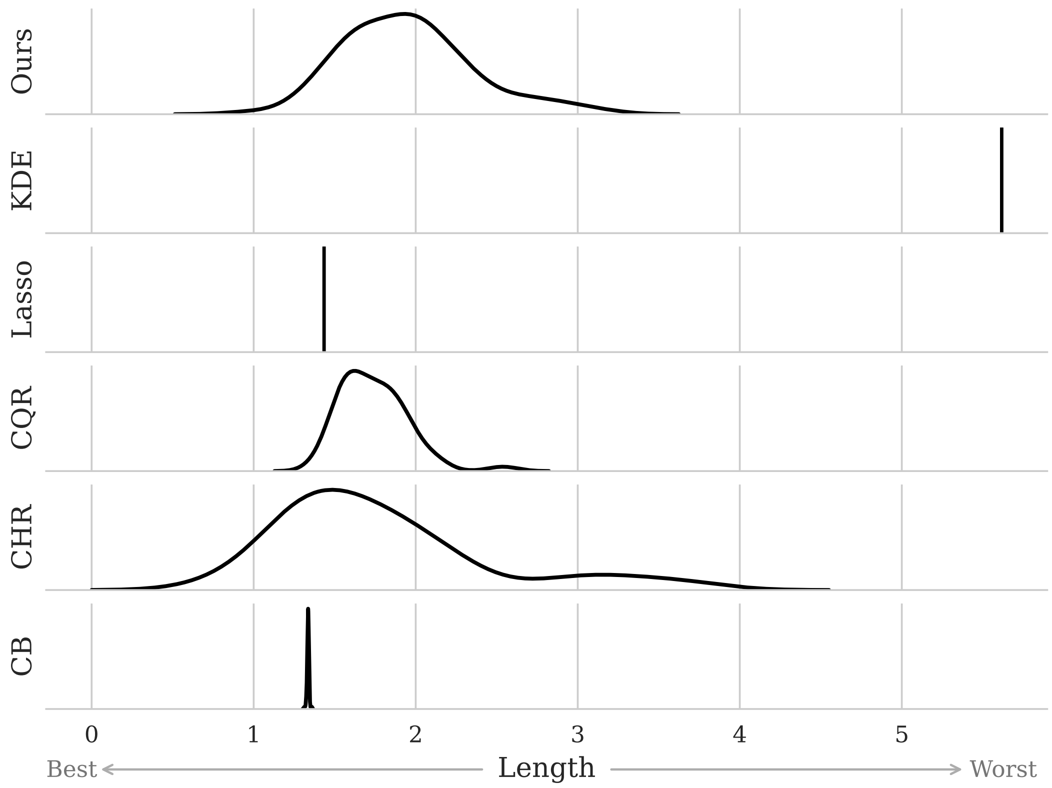
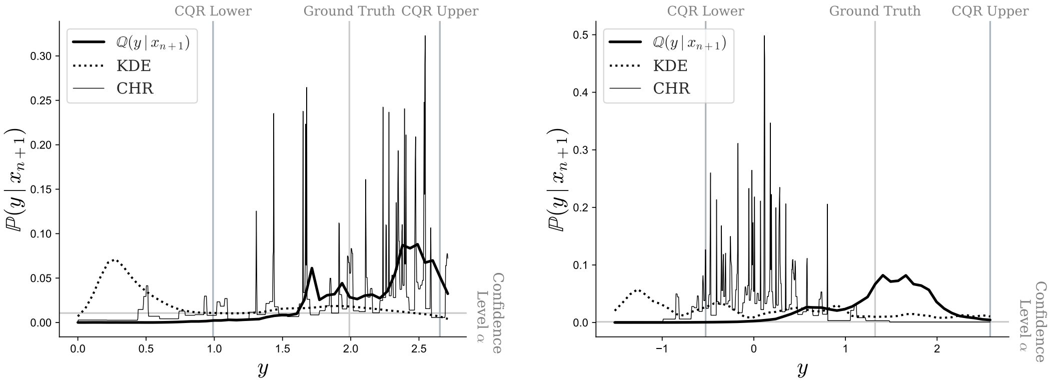


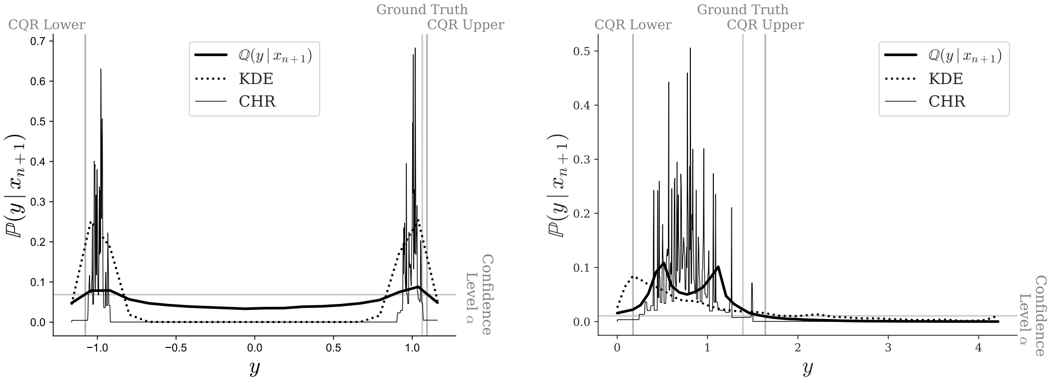
Figure 19: Stock Length Distribution across all CP Methods



(a) BIO

(b) BREAST CANCER

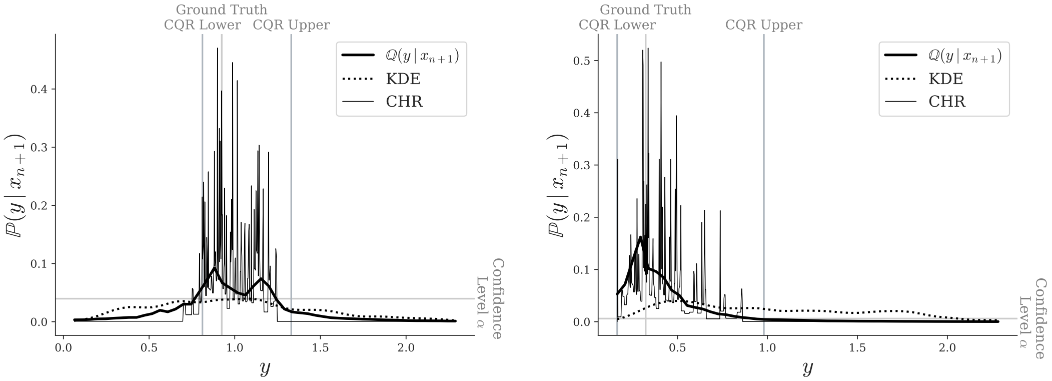
Figure 20: Plot of outputted density functions for ours, KDE, and CHR on for BIO and BREAST CANCER



(a) BIMODAL

(b) COMMUNITY

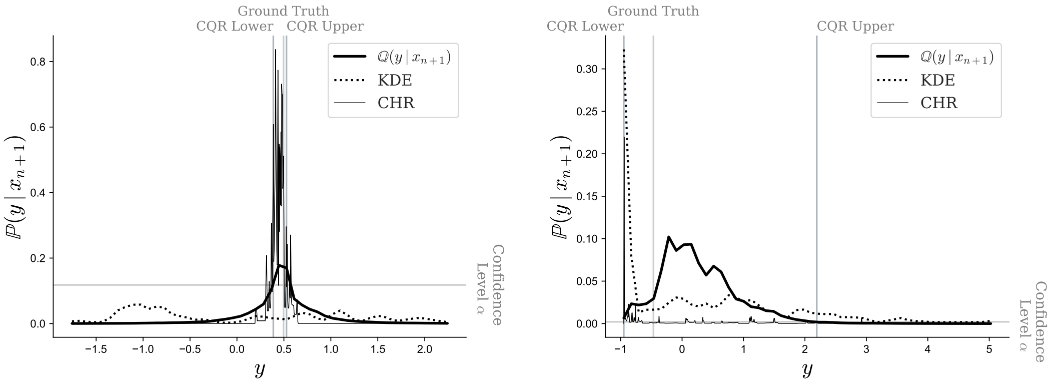
Figure 21: Plot of outputted density functions for ours, KDE, and CHR on for BIMODAL and COMMUNITY



(a) CONCRETE

(b) DIABETES

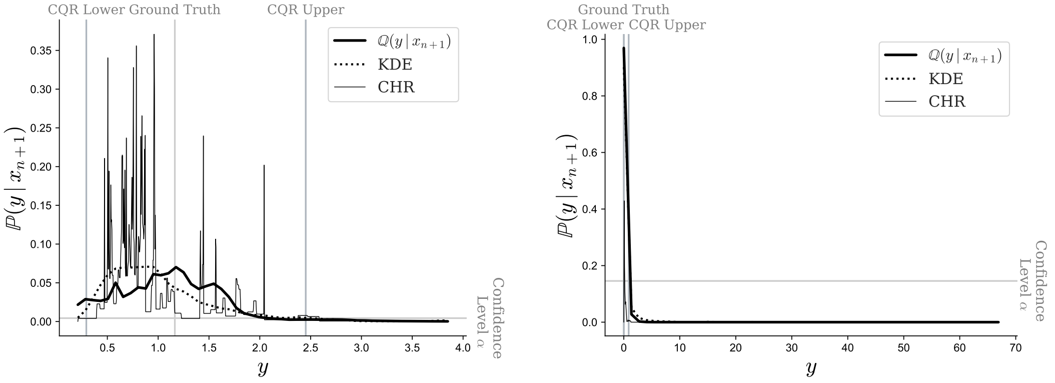
Figure 22: Plot of outputted density functions for ours, KDE, and CHR on for CONCRETE and DIABETES



(a) ENERGY

(b) FOREST

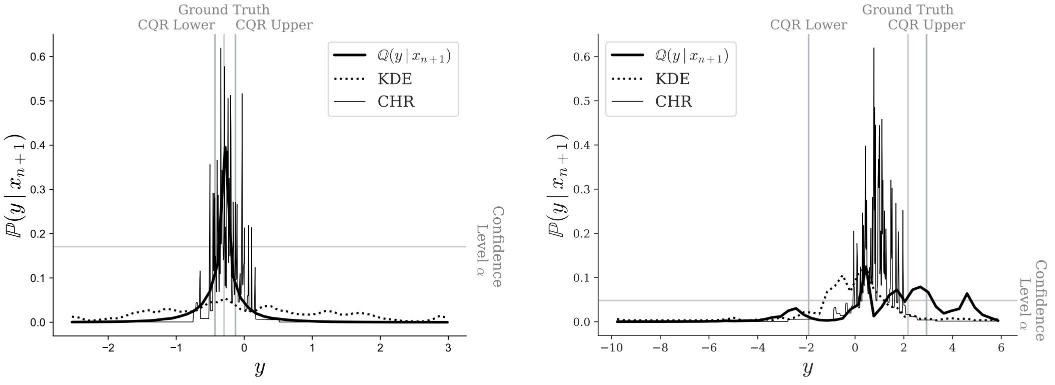
Figure 23: Plot of outputted density functions for ours, KDE, and CHR on for ENERGY and FOREST



(a) LOG NORMAL

(b) MEPS-21

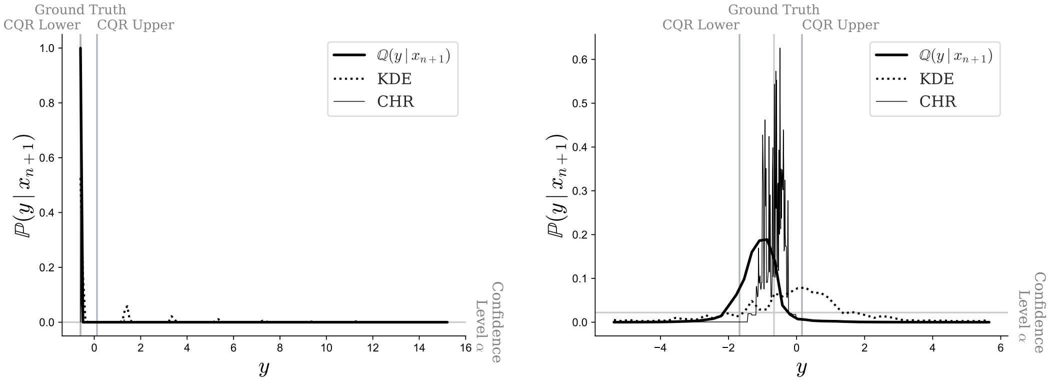
Figure 24: Plot of outputted density functions for ours, KDE, and CHR on for LOG NORMAL and MEPS-19



(a) PARKINSON'S

(b) PENDULUM

Figure 25: Plot of outputted density functions for ours, KDE, and CHR on for PARKINSON'S and PENDULUM



(a) SOLAR

(b) STOCK

Figure 26: Plot of outputted density functions for ours, KDE, and CHR on for SOLAR and STOCK