Conformal Inference is (almost) Free for Neural Networks Trained with Early Stopping

Ziyi Liang [*1] **Yanfei Zhou** [*2] **Matteo Sesia** [2]

Abstract

Early stopping based on hold-out data is a popular regularization technique designed to mitigate overfitting and increase the predictive accuracy of neural networks. Models trained with early stopping often provide relatively accurate predictions, but they generally still lack precise statistical guarantees unless they are further calibrated using independent hold-out data. This paper addresses the above limitation with *conformalized early stopping*: a novel method that combines early stopping with conformal calibration while efficiently recycling the same hold-out data. This leads to models that are both accurate and able to provide exact predictive inferences without multiple data splits nor overly conservative adjustments. Practical implementations are developed for different learning tasks—outlier detection, multi-class classification, regression—and their competitive performance is demonstrated on real data.

1. Introduction

Deep neural networks can detect complex data patterns and leverage them to make accurate predictions in many applications, including computer vision, natural language processing, and speech recognition, to name a few examples. These models can sometimes even outperform skilled humans (Silver et al., 2016), but they still make mistakes. Unfortunately, the severity of these mistakes is compounded by the fact that the predictions computed by neural networks are often overconfident (Guo et al., 2017), partly due to overfitting (Thulasidasan et al., 2019; Ovadia et al., 2019). Several training strategies have been developed to mitigate overfitting, including dropout (Srivastava et al., 2014), batch normalization (Ioffe & Szegedy, 2015), weight normalization (Salimans & Kingma, 2016), data augmentation (Shorten & Khoshgoftaar, 2019), and early stopping (Prechelt, 1998); the latter is the focus of this paper.

Early stopping consists of continuously evaluating after each batch of stochastic gradient updates (or *epoch*) the predictive performance of the current model on *hold-out* independent data. After a large number of gradient updates, only the intermediate model achieving the best performance on the hold-out data is utilized to make predictions. This strategy is often effective at mitigating overfitting and can produce relatively accurate predictions compared to fully trained models, but it does not fully resolve overconfidence because it does not lead to models with finite-sample guarantees.

A general framework for quantifying the predictive uncertainty of any *black-box* machine learning model is that of conformal inference (Vovk et al., 2005). The key idea of conformal inference is to apply a pre-trained model to a *calibration* set of hold-out observations drawn at random from the target population. If the calibration data are exchangeable with the test point of interest, the model performance on the calibration set can be translated into statistically rigorous predictive inferences. This framework is flexible and can accommodate different learning tasks, including out-of-distribution testing (Smith et al., 2015), classification (Vovk et al., 2003), and regression (Vovk et al., 2005). For example, in the context of classification, conformal inference can give prediction sets that contain the correct label for a new data point with high probability. In theory, the quality of the trained model has no consequence on the *average* validity of conformal inferences, but it does affect their reliability and usefulness on a case-by-case level. In particular, conformal uncertainty estimates obtained after calibrating an overconfident model may be too conservative for some test cases and too optimistic for others (Romano et al., 2020b). The goal of this paper is to combine conformal calibration with standard early stopping training techniques as efficiently as possible, in order to produce more reliable predictive inferences with a finite amount of available data.

Achieving high accuracy with deep learning often requires large training sets (Marcus, 2018), and conformal inference

[*]Equal contribution [1]Department of Mathematics, University of Southern California, Los Angeles, CA, USA [2]Department of Data Sciences and Operations, University of Southern California, Los Angeles, CA, USA. Correspondence to: Matteo Sesia <sesia@marshall.usc.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

makes the overall pipeline even more data-intensive. As high-quality observations can be expensive to collect, in some situations practitioners may naturally wonder whether the advantage of having principled uncertainty estimates is worth a possible reduction in predictive accuracy due to fewer available training samples. This concern is relevant because the size of the calibration set cannot be too small if one wants stable and reliable conformal inferences (Vovk, 2012; Sesia & Candès, 2020). In fact, very large calibration sets may be necessary to obtain stronger conformal inferences that are valid not only on average but also conditionally on some important individual features; see Vovk et al. (2003); Romano et al. (2020a); Barber et al. (2021a).

This paper resolves the above dilemma by showing that conformal inferences for deep learning models trained with early stopping can be obtained almost "for free"—without spending precious data. More precisely, we present an innovative method that blends model training with early stopping and conformal calibration using the same hold-out samples, essentially obtaining rigorous predictive inferences at no additional data cost compared to standard early stopping. It is worth emphasizing this result is not trivial. In fact, naively applying existing conformal calibration methods using the same hold-out samples utilized for early stopping would not lead to theoretically valid inferences, at least not without resorting to very conservative corrections.

The paper is organized as follows. Section 2 develops our *conformalized early stopping* (CES) method, starting from outlier detection and classification, then addressing regression. Section 3 demonstrates the advantages of CES through numerical experiments. Section 4 concludes with a discussion and some ideas for further research. Additional details and results, including a theoretical analysis of the naive benchmark mentioned above, can be found in the Appendices, along with all mathematical proofs.

Related Work

Conformal inference (Saunders et al., 1999; Vovk et al., 1999; 2005) has become a very rich and active area of research (Lei et al., 2013; Lei & Wasserman, 2014; Lei et al., 2018b; Barber et al., 2021b). Many prior works studied the computation of efficient conformal inferences starting from pre-trained *black-box* models, including for example in the context of outlier detection (Smith et al., 2015; Guan & Tibshirani, 2022; Liang et al., 2022; Bates et al., 2023), classification (Vovk et al., 2003; Hechtlinger et al., 2018; Romano et al., 2020b; Angelopoulos et al., 2021; Bates et al., 2021), and regression (Vovk et al., 2005; Lei & Wasserman, 2014; Romano et al., 2019). Other works have studied the general robustness of conformal inferences to distribution shifts (Tibshirani et al., 2019; Sesia et al., 2022) and, more broadly, to failures of the data exchangeability assumption (Barber et al., 2022; Gibbs & Candès, 2022). Our research is orthogonal, as we look inside the black-box model and develop a novel early-stopping training technique that is naturally integrated with conformal calibration. Nonetheless, the proposed method could be combined with those described in the aforementioned papers. Other recent research has explored different ways of bringing conformal inference into the learning algorithms (Colombo & Vovk, 2020; Bellotti, 2021; Stutz et al., 2021; Einbinder et al., 2022), and some of those works apply standard early stopping techniques, but they do not address our problem.

This paper is related to Yang & Kuchibhotla (2021), which proposed a general theoretical adjustment for conformal inferences computed after model selection. That method could be utilized to account for early stopping without further data splits, as detailed in Appendix A1. However, we will demonstrate that even an improved version of such analysis remains overly conservative in the context of model selection via early stopping, and the alternative method developed in this paper performs much better in practice. Our solution is inspired by Mondrian conformal inference (Vovk et al., 2003) as well as by the integrative conformal method of Liang et al. (2022). The latter deals with the problem of selecting the best model from an arbitrary machine learning toolbox to obtain the most powerful conformal p-values for outlier testing. The idea of Liang et al. (2022) extends naturally to the early stopping problem in the special cases of outlier detection and classification, but the regression setting requires substantial technical innovations. The work of Liang et al. (2022) is also related to Marandon et al. (2022), although the latter is more distant from this paper because it focuses on theoretically controlling the false discovery rate (Benjamini & Hochberg, 1995) in multiple testing problems. Finally, this paper draws inspiration from Kim et al. (2020), which shows that machine learning models trained with bootstrap (or bagging) techniques can also lead to valid conformal inferences essentially for free.

2. Methods

2.1. Standard Conformal Inference and Early Stopping

Consider n data points, Z_i for $i \in \mathcal{D} = [n] = \{1, \ldots, n\}$, sampled exchangeably (e.g., i.i.d.) from an unknown distribution P_Z with support on some space $\mathcal{Z}$. Consider also an additional test sample, Z_{n+1}. In the context of outlier detection, one wishes to test whether Z_{n+1} was sampled exchangeably from P_Z. In classification or regression, one can write $Z_i = (X_i, Y_i)$, where X_i is a feature vector while Y_i is a discrete category or a continuous response, and the goal is to predict the unobserved value of Y_{n+1} given X_{n+1} and the data in $\mathcal{D}$.

The standard pipeline begins by randomly splitting the data in $\mathcal{D}$ into three disjoint subsets: $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{es}}, \mathcal{D}_{\text{cal}} \subset [n]$. The samples in $\mathcal{D}_{\text{train}}$ are utilized to train a model M via stochastic gradient descent, in such a way as to (approximately) minimize the desired loss $\mathcal{L}$, while the observations in $\mathcal{D}_{\text{es}}$ and $\mathcal{D}_{\text{cal}}$ are held out. We denote by M_t the model learnt after t epochs of stochastic gradient descent, for any $t \in [t^{\text{max}}]$, where t^{max} is a pre-determined maximum number of epochs. For simplicity, $\mathcal{L}$ is assumed to be an additive loss, in the sense that its value calculated on the training data after t epochs is $\mathcal{L}_{\text{train}}(M_t) = \sum_{i \in \mathcal{D}_{\text{train}}} \ell(M_t; Z_i)$, for some appropriate function ℓ. For example, a typical choice for regression would be the squared-error loss: $\ell(M_t; Z_i) = [Y_i - \hat{\mu}(X_i; M_t)]^2$, where $\hat{\mu}(X_i; M_t)$ indicates the value of the regression function at X_i, as estimated by M_t. Similarly, the loss evaluated on $\mathcal{D}_{\text{es}}$ is denoted as $\mathcal{L}_{\text{es}}(M_t) = \sum_{i \in \mathcal{D}_{\text{es}}} \ell(M_t; Z_i)$. After training for t^{max} epochs, early stopping selects the model $\hat{M}_{\text{es}}$ that minimizes the loss on $\mathcal{D}_{\text{es}}$: $\hat{M}_{\text{es}} = \arg\min_{M_t \,:\, 0 \le t \le t^{\text{max}}} \mathcal{L}_{\text{es}}(M_t)$. Conformal calibration of $\hat{M}_{\text{es}}$ is then conducted using the independent hold-out data set $\mathcal{D}_{\text{cal}}$, as sketched in Figure 1 (a). This pipeline requires a three-way data split because: (i) $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es}}$ must be disjoint to ensure the early stopping criterion is effective at mitigating overfitting; and (ii) $\mathcal{D}_{\text{cal}}$ must be disjoint from $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{es}}$ to ensure the performance of the selected model $\hat{M}_{\text{es}}$ on the calibration data gives us an unbiased preview of its future performance at test time.

2.2. Preview of our Contribution

This paper develops a novel method to jointly carry out both early stopping and conformal inference using a single hold-out data set, denoted in the following as $\mathcal{D}_{\text{es-cal}}$. The advantage of this approach is that it allows more samples to be allocated to $\mathcal{D}_{\text{train}}$. This is not a straightforward problem. For example, one cannot naively apply standard conformal inference methods using the same hold-out set $\mathcal{D}_{\text{es-cal}}$ previously used for early stopping, as detailed in Appendix A1. In that case, the early stopping decision would invalidate the conformal inferences by breaking the exchangeability between the calibration data and the test point, as the latter is not used to select the predictive model. As explained in Appendix A1.2, it is possible to correct conformal inferences obtained with this naive approach by adjusting the nominal coverage level conservatively, leveraging suitable concentration inequalities (Yang & Kuchibhotla, 2021). However, such theoretical corrections tend to be overly pessimistic in practice and may often be too conservative to be useful; this is demonstrated by the numerical experiments described in Section 3 and previewed here in Figure 2.

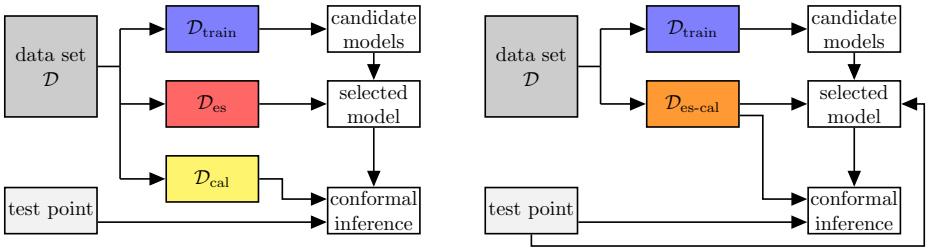
By contrast, the CES method proposed in this paper is based on the following idea inspired by Liang et al. (2022). Valid conformal inferences can be obtained by calibrating $\hat{M}_{\text{es}}$ using the same data set $\mathcal{D}_{\text{es-cal}}$ used for model selection, as long as the test sample Z_{n+1} is also involved in the early stopping rule exchangeably with all other samples in $\mathcal{D}_{\text{es-cal}}$. This concept, illustrated schematically in Figure 1 (b), is not obvious to translate into a practical method, however, for two reasons. First, the ground truth for the test point (i.e., its outlier status or its outcome label) is unknown. Second, the method may need to be repeatedly applied for a large number of distinct test points in a computationally efficient way, and one cannot re-train the model separately for each test point. In the next section, we will explain how to overcome these challenges in the special case of early stopping for outlier detection; then, the solution will be extended to the classification and regression settings.

2.3. CES for Outlier Detection

Consider testing whether Z_{n+1}, is an *inlier*, in the sense that it was sampled from P_Z exchangeably with the data in $\mathcal{D}$. Following the notation of Section 2.1, consider a partition of $\mathcal{D}$ into two subsets, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$, chosen at random independently of everything else, such that $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{es-cal}}$. The first step of CES consists of training a deep one-class classifier M using the data in $\mathcal{D}_{\text{train}}$ via stochastic gradient descent for t^{max} epochs, storing all parameters characterizing the intermediate model after each τ epochs. We refer to $\tau \in [t^{\text{max}}]$ as the *storage period*, a parameter pre-defined by the user. Intuitively, a smaller τ increases the memory cost of CES but may also lead to the selection of a more accurate model. While the memory cost of this approach is higher compared to that of standard early-stopping training techniques, which only require storing one model at a time, it is not prohibitively expensive. In fact, the candidate models do not need to be kept in precious RAM memory but can be stored on a relatively cheap hard drive. As reasonable choices of τ may typically be in the order of $T = \lfloor t^{\text{max}}/\tau \rfloor \approx 100$, the cost of CES is not excessive in many real-world situations. For example, it takes approximately 100 MB to store a pre-trained standard ResNet50 computer vision model, implying that CES would require approximately 10 GB of storage in such applications—today this costs less than \$0.25/month in the cloud.

After pre-training and storing T candidate models, namely $M_{t_1}, \ldots, M_{t_T}$ for some sub-sequence $(t_1, \ldots, t_T)$ of $[t^{\text{max}}]$, the next step is to select the appropriate early-stopped model based on the hold-out data in $\mathcal{D}_{\text{es-cal}}$ as well as the test point Z_{n+1}. Following the notation of Section 2.1, define the value of the one-class classification loss $\mathcal{L}$ for model M_t, for any $t \in [T]$, evaluated on $\mathcal{D}_{\text{es-cal}}$ as: $\mathcal{L}_{\text{es-cal}}(M_t) = \sum_{i \in \mathcal{D}_{\text{es-cal}}} \ell(M_t; Z_i)$. Further, for any $z \in \mathcal{Z}$, define also $\mathcal{L}^{+1}_{\text{es-cal}}(M_t, z)$ as:

$$\mathcal{L}^{+1}_{\text{es-cal}}(M_t, z) = \mathcal{L}_{\text{es-cal}}(M_t) + \ell(M_t; z). \tag{1}$$



(a) Conformal inference after early stopping. (b) Conformalized early stopping (CES).

Figure 1. Conformal inference for models trained with early stopping. (a) Conventional pipeline requiring a three-way sample split. (b) Conformalized early stopping, requiring a two-way split.

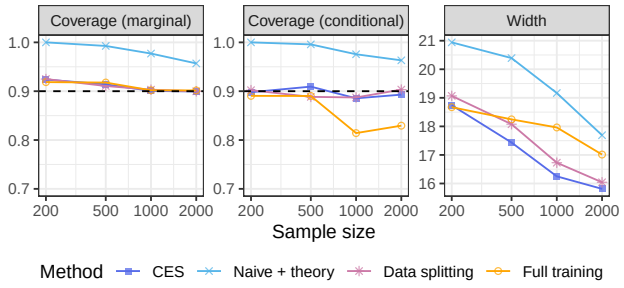


Figure 2. Average performance, as a function of the sample size, of conformal inferences based on neural networks trained and calibrated with different methods, on the *bio* regression data (dat, b). Ideally, the coverage of the conformal prediction intervals should be close to 90% and their width should be small. All methods shown here guarantee 90% marginal coverage. See Table A3 for more detailed results and standard errors.

Therefore, $\mathcal{L}^{+1}_{\text{es-cal}}(M_t, Z_{n+1})$ can be interpreted as the cumulative value of the loss function calculated on an augmented hold-out data set including also Z_{n+1}. Then, we select the model $\hat{M}_{\text{ces}}(Z_{n+1})$ minimizing $\mathcal{L}^{+1}_{\text{es-cal}}(M_t, Z_{n+1})$:

$$\hat{M}_{\text{ces}}(Z_{n+1}) = \underset{M_{t_j}\,:\,1\leq j\leq T}{\arg\min}\ \mathcal{L}^{+1}_{\text{es-cal}}(M_{t_j}, Z_{n+1}). \quad (2)$$

Note that the computational cost of evaluating (2) is negligible compared to that of training the models.

Next, the selected model $\hat{M}_{\text{ces}}(Z_{n+1})$ is utilized to compute a conformal p-value (Bates et al., 2023) to test whether Z_{n+1} is an inlier. In particular, $\hat{M}_{\text{ces}}(Z_{n+1})$ is utilized to compute *nonconformity scores* $\hat{S}_i(Z_{n+1})$ for all samples $i \in \mathcal{D}_{\text{es-cal}} \cup \{n+1\}$. These scores rank the observations in $\mathcal{D}_{\text{es-cal}} \cup \{n+1\}$ based on how the one-class classifier $\hat{M}_{\text{ces}}(Z_{n+1})$ perceives them to be similar to the training data; by convention, a smaller value of $\hat{S}_i(Z_{n+1})$ suggests Z_i is more likely to be an outlier. Suitable scores are typically included in the output of standard one-class classification models, such as those provided by the Python library PyTorch. For simplicity, we assume all scores are almost-surely distinct; otherwise, ties can be broken at random by adding a small amount of independent noise. Then, the conformal p-value $\hat{u}_0(Z_{n+1})$ is given by the usual formula:

$$\hat{u}_0(Z_{n+1}) = \frac{1 + |i \in \mathcal{D}_{\text{es-cal}} : \hat{S}_i \leq \hat{S}_{n+1}|}{1 + |\mathcal{D}_{\text{es-cal}}|}, \quad (3)$$

making the dependence of $\hat{S}_i$ on Z_{n+1} implicit in the interest of space. This method, outlined by Algorithm A4 in Appendix A2, gives p-values that are exactly valid in finite samples, in the sense that they are stochastically dominated by the uniform distribution under the null hypothesis.

Theorem 2.1. *Assume $Z_1, \ldots, Z_n, Z_{n+1}$ are exchangeable random samples, and let $\hat{u}_0(Z_{n+1})$ be the output of Algorithm A4, as given in (3). Then, $\mathbb{P}\left[\hat{u}_0(Z_{n+1}) \leq \alpha\right] \leq \alpha$ for any $\alpha \in (0,1)$.*

2.4. CES for Classification

The above CES method will now be extended to deal with K-class classification problems, for any $K \geq 2$. Consider n exchangeable pairs of observations (X_i, Y_i), for $i \in \mathcal{D} = [n]$, and a test point (X_{n+1}, Y_{n+1}) whose label $Y_{n+1} \in [K]$ has not yet been observed. The goal is to construct an informative prediction set for Y_{n+1} given the observed features X_{n+1} and the rest of the data, assuming (X_{n+1}, Y_{n+1}) is exchangeable with the observations indexed by $\mathcal{D}$. An ideal goal would be to construct the smallest possible prediction set with guaranteed *feature-conditional coverage* at level $1-\alpha$, for any fixed $\alpha \in (0,1)$. Formally, a prediction set $\hat{C}_\alpha(X_{n+1}) \subseteq [K]$ has feature-conditional coverage at level $1-\alpha$ if $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1}) \mid X_{n+1} = x] \geq 1-\alpha$, for any $x \in \mathcal{X}$, where $\mathcal{X}$ is the feature space. Unfortunately, perfect feature-conditional coverage is extremely difficult to achieve unless the feature space $\mathcal{X}$ is very small. Therefore, in practice, one must be satisfied with obtaining relatively weaker guarantees, such as *label-conditional coverage* and *marginal coverage*. Formally, $\hat{C}_\alpha(X_{n+1})$ has $1-\alpha$ label-conditional coverage if $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1}) \mid Y_{n+1} = y] \geq 1-\alpha$, for any $y \in [K]$, while marginal coverage corresponds to $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})] \geq 1-\alpha$. Label-conditional cover-

age is stronger than marginal coverage, but both criteria are useful because the latter is easier to achieve with smaller (and hence more informative) prediction sets.

We begin by focusing on label-conditional coverage, as this follows most easily from the results of Section 2.3. This solution will be extended in Appendix A3 to target marginal coverage. The first step of CES consists of randomly splitting $\mathcal{D}$ into two subsets, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$, as in Section 2.3. The samples in $\mathcal{D}_{\text{es-cal}}$ are further divided into subsets $\mathcal{D}^y_{\text{es-cal}}$ with homogeneous labels; that is, $\mathcal{D}^y_{\text{es-cal}} = \{i \in \mathcal{D}_{\text{es-cal}} : Y_i = y\}$ for each $y \in [K]$. The data in $\mathcal{D}_{\text{train}}$ are utilized to train a neural network classifier via stochastic gradient descent, storing the intermediate candidate models M_t after each τ epochs. This is essentially the same approach as in Section 2.3, with the only difference being that the neural network is now designed to perform K-class classification rather than one-class classification. Therefore, this neural network should have a soft-max layer with K nodes near its output, whose values corresponding to an input data point with features x are denoted as $\hat{\pi}_y(x)$, for all $y \in [K]$. Intuitively, we will interpret $\hat{\pi}_y(x)$ as approximating (possibly inaccurately) the true conditional data-generating distribution; i.e., $\hat{\pi}_y(x) \approx \mathbb{P}[Y = y \mid X = x]$.

For any model M_t, any $x \in \mathcal{X}$, and any $y \in [K]$, define the augmented loss $\mathcal{L}^{+1}_{\text{es-cal}}(M_t, x, y)$ as:

$$\mathcal{L}^{+1}_{\text{es-cal}}(M_t, x, y) = \mathcal{L}_{\text{es-cal}}(M_t) + \ell(M_t; x, y). \quad (4)$$

Concretely, a typical choice for ℓ is the cross-entropy loss: $\ell(M_t; x, y) = -\log \hat{\pi}^t_y(x)$, where $\hat{\pi}^t$ denotes the soft-max probability distribution estimated by model M_t. Intuitively, $\mathcal{L}^{+1}_{\text{es-cal}}(M_t, x, y)$ is the cumulative value of the loss function calculated on an augmented hold-out data set including also the imaginary test sample (x, y). Then, for any $y \in [K]$, CES selects the model $\hat{M}_{\text{ces}}(X_{n+1}, y)$ minimizing $\mathcal{L}^{+1}_{\text{es-cal}}(M_t, X_{n+1}, y)$ among the T stored models:

$$\hat{M}_{\text{ces}}(X_{n+1}, y) = \underset{M_{t_j} : 1 \le j \le T}{\arg\min} \mathcal{L}^{+1}_{\text{es-cal}}(M_{t_j}, X_{n+1}, y). \quad (5)$$

The selected model $\hat{M}_{\text{ces}}(X_{n+1}, y)$ is then utilized to compute a conformal p-value for testing whether $Y_{n+1} = y$. In particular, we compute nonconformity scores $\hat{S}^y_i(X_{n+1})$ for all $i \in \mathcal{D}^y_{\text{es-cal}} \cup \{n+1\}$, imagining that $Y_{n+1} = y$. Different types of nonconformity scores can be easily accommodated, but in this paper, we follow the *adaptive* strategy of Romano et al. (2020b). The computation of these nonconformity scores based on the selected model $\hat{M}_{\text{ces}}$ is reviewed in Appendix A4. Here, we simply note the p-value is given by:

$$\hat{u}_y(X_{n+1}) = \frac{1 + |i \in \mathcal{D}^y_{\text{es-cal}} : \hat{S}^y_i \le \hat{S}^y_{n+1}|}{1 + |\mathcal{D}^y_{\text{es-cal}}|}, \quad (6)$$

again making the dependence of $\hat{S}^y_i$ on X_{n+1} implicit. Finally, the prediction set $\hat{C}_\alpha(X_{n+1})$ is constructed by including all possible labels for which the corresponding null hypothesis cannot be rejected at level α:

$$\hat{C}_\alpha(X_{n+1}) = \{y \in [K] : \hat{u}_y(X_{n+1}) \ge \alpha\}. \quad (7)$$

This method, outlined by Algorithm A5 in Appendix A2, guarantees label-conditional coverage at level $1 - \alpha$.

Theorem 2.2. *Assume $(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})$ are exchangeable, and let $\hat{C}_\alpha(X_{n+1})$ be the output of Algorithm A5, as given in (7), for any given $\alpha \in (0, 1)$. Then, $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1}) \mid Y_{n+1} = y] \ge 1 - \alpha$ for any $y \in [K]$.*

2.5. CES for Regression

This section extends CES to regression problems with a continuous outcome. As in the previous sections, consider a data set containing n exchangeable observations (X_i, Y_i), for $i \in \mathcal{D} = [n]$, and a test point (X_{n+1}, Y_{n+1}) with a latent label $Y_{n+1} \in \mathbb{R}$. The goal is to construct a reasonably narrow *prediction interval* $\hat{C}_\alpha(X_{n+1})$ for Y_{n+1} that is guaranteed to have marginal coverage above some level $1 - \alpha$, i.e., $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})] \ge 1 - \alpha$, and can also practically achieve reasonably high feature-conditional coverage. Developing a CES method for this problem is more difficult compared to the classification case studied in Section 2.4 due to the infinite number of possible values for Y_{n+1}. In fact, a naive extension of Algorithm A5 would be computationally unfeasible in the regression setting, for the same reason why full-conformal prediction (Vovk et al., 2005) is generally impractical. The novel solution described below is designed to leverage the particular structure of an early stopping criterion based on the squared-error loss evaluated on hold-out data. Focusing on the squared-error loss makes CES easier to implement and explain using classical *absolute residual* nonconformity scores (Vovk et al., 2005; Lei et al., 2018a). However, similar ideas could also be repurposed to accommodate other scores, such as those based on quantile regression (Romano et al., 2019), conditional distributions (Chernozhukov et al., 2021; Izbicki et al., 2020), or conditional histograms (Sesia & Romano, 2021).

As usual, we randomly split $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$. The data in $\mathcal{D}_{\text{train}}$ are utilized to train a neural network via stochastic gradient descent, storing the intermediate models M_t after each τ epoch. The approach is similar to those in Sections 2.3–2.4, although now the output of a model M_t applied to a sample with features x is denoted by $\hat{\mu}_t(x)$ and is designed to approximate (possibly inaccurately) the conditional mean of the unknown data-generating distribution; i.e., $\hat{\mu}_t(x) \approx \mathbb{E}[Y \mid X = x]$. (Note that we will omit the superscript t unless necessary to avoid ambiguity). For any

model M_t and any $x \in \mathcal{X}$, $y \in \mathbb{R}$, define

$$\mathcal{L}^{+1}_{\text{es-cal}}(M_t, x, y) = \mathcal{L}_{\text{es-cal}}(M_t) + [y - \hat{\mu}_t(x)]^2. \quad (8)$$

Consider now the following optimization problem,

$$\hat{M}_{\text{ces}}(X_{n+1}, y) = \underset{M_{t_j}\,:\,1 \le j \le T}{\arg\min} \mathcal{L}^{+1}_{\text{es-cal}}(M_{t_j}, X_{n+1}, y), \quad (9)$$

which can be solved simultaneously for all $y \in \mathbb{R}$ thanks to the amenable form of (8). In fact, each $\mathcal{L}^{+1}_{\text{es-cal}}(M_t, x, y)$ is a simple quadratic function of y; see the sketch in Figure 3. This implies $\hat{M}_{\text{ces}}(X_{n+1}, y)$ is a step function, whose parameters can be computed at cost $\mathcal{O}(T \log T)$ with an efficient divide-and-conquer algorithm designed to find the lower envelope of a family of parabolas (Devillers & Golin, 1995; Nielsen & Yvinec, 1998); see Appendix A5.

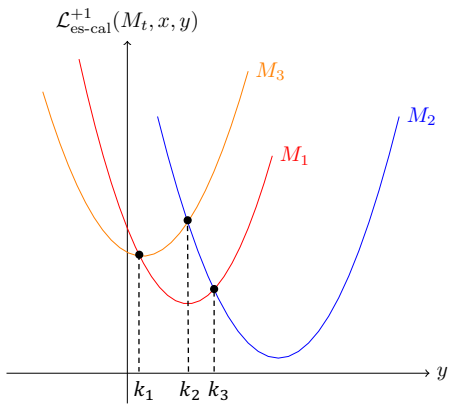


Figure 3. Squared-error loss on test-augmented hold-out data for three alternative regression models M_1, M_2 and M_3, as a function of the place-holder outcome y for the test point. The CES method utilizes the best model for each possible value of y, which is identified by the lower envelope of these three parabolas. In this case, the lower envelope has two finite knots at k_1 and k_3.

Therefore, $\hat{M}_{\text{ces}}(X_{n+1}, y)$ has L distinct steps, for some $L = \mathcal{O}(T \log T)$ that may depend on X_{n+1}, and it can be written as a function of y as:

$$\hat{M}_{\text{ces}}(X_{n+1}, y) = \sum_{l=1}^{L} m_l(X_{n+1}) \mathbb{1}\left[y \in (k_{l-1}, k_l]\right], \quad (10)$$

where $m_l(X_{n+1}) \in [T]$ represents the best model selected within the interval $(k_{l-1}, k_l]$ such that $m_l(X_{n+1}) \neq m_{l-1}(X_{n+1})$ for all $l \in [L]$. Above, $k_1 \le k_2 \le \cdots \le k_L$ denote the *knots* of $\hat{M}_{\text{ces}}(X_{n+1}, y)$, which also depend on X_{n+1} and are defined as the boundaries in the domain of y between each consecutive pair of steps, with the understanding that $k_0 = -\infty$ and $k_{L+1} = +\infty$. Then, for each step $l \in [L]$, let $\mathcal{B}_l$ indicate the interval $\mathcal{B}_l = (k_{l-1}, k_l]$ and, for all $i \in \mathcal{D}_{\text{es-cal}}$, evaluate the nonconformity score $\hat{S}_i(X_{n+1}, \mathcal{B}_l)$ for observation (X_i, Y_i) based on the regression model indicated by $m_l(X_{n+1})$; i.e.,

$$\hat{S}_i(X_{n+1}, \mathcal{B}_l) = |Y_i - \hat{\mu}_{m_l(X_{n+1})}(X_i)|. \quad (11)$$

Let $\hat{Q}_{1-\alpha}(X_{n+1}, \mathcal{B}_l)$ denote the $\lceil (1-\alpha)(1 + |\mathcal{D}_{\text{es-cal}}|) \rceil$-th smallest value among all nonconformity scores $\hat{S}_i(X_{n+1}, \mathcal{B}_l)$, assuming for simplicity that there are no ties; otherwise, ties can be broken at random. Then, define the interval $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)$ as that obtained by applying the standard conformal prediction method with absolute residual scores based on the regression model $\hat{\mu}_{m_l(X_{n+1})}(X_{n+1})$:

$$\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l) = \hat{\mu}_{m_l(X_{n+1})}(X_{n+1}) \pm \hat{Q}_{1-\alpha}(X_{n+1}, \mathcal{B}_l). \quad (12)$$

Finally, the prediction interval $\hat{C}_\alpha(X_{n+1})$ is given by:

$$\hat{C}_\alpha(X_{n+1}) = \text{Convex}\left(\cup_{l=1}^{L} \{\mathcal{B}_l \cap \hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)\}\right), \quad (13)$$

where $\text{Convex}(\cdot)$ denotes the convex hull of a set. This procedure is summarized in Algorithm A6 and it is guaranteed to produce prediction sets with valid marginal coverage.

Theorem 2.3. *Assume $(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})$ are exchangeable, and let $\hat{C}_\alpha(X_{n+1})$ be the output of Algorithm A6, as given by (13), for any given $\alpha \in (0, 1)$. Then, $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})] \ge 1 - \alpha$.*

The intuition behind the above method is as follows. Each intermediate interval $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)$, for $l \in [L]$, may be thought of as being computed by applying, under the null hypothesis that $Y_{n+1} \in \mathcal{B}_l$, the classification method from Section 2.4 for a discretized version of our problem based on the partition $\{\mathcal{B}_l\}_{l=1}^{L}$. Then, leveraging the classical duality between confidence intervals and p-values, it becomes clear that taking the intersection of $\mathcal{B}_l$ and $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)$ essentially amounts to including the "label" $\mathcal{B}_l$ in the output prediction if the null hypothesis $Y_{n+1} \in \mathcal{B}_l$ cannot be rejected. The purpose of the final convex hull operation is to generate a contiguous prediction interval, which is what we originally stated to seek.

One may intuitively be concerned that this method may output excessively wide prediction interval if the location of $\{\mathcal{B}_l \cap \hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)\}$ is extremely large in absolute value. However, our numerical experiments will demonstrate that, as long as the number of calibration data points is not too small, the selected models in general provide reasonably concentrated predictions around the true test response regardless of the placeholder value y. Therefore, the interval $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)$ tends to be close to the true y even if $\mathcal{B}_l$ is far away, in which case $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l) \cap \mathcal{B}_l = \emptyset$ does not expand the final prediction interval $\hat{C}_\alpha(X_{n+1})$.

Although it is unlikely, Algorithm A6 may sometimes produce an empty set, which is an uninformative and potentially confusing output. A simple solution consists of replacing any empty output with the naive conformal prediction interval computed by Algorithm A3 in Appendix A1, which leverages an early-stopped model selected by looking at the original calibration data set without the test point. This approach is outlined by Algorithm A10 in Appendix A6. As the intervals given by Algorithm A10 always contain those output by Algorithm A6, it follows that Algorithm A10 also enjoys guaranteed coverage; see Corollary A1.

2.6. CES for Quantile Regression

The CES method for regression described in Section 2.5 relies on classical nonconformity scores (Vovk et al., 2005; Lei et al., 2018a) that are not designed to deal efficiently with heteroscedastic data (Romano et al., 2019; Sesia & Candès, 2020). However, the idea can be extended to accommodate other nonconformity scores, including those based on quantile regression (Romano et al., 2019), conditional distributions (Izbicki et al., 2020; Chernozhukov et al., 2021), or conditional histograms (Sesia & Romano, 2021). The reason why we have so far focused on the classical absolute residual scores is that they are more intuitive to apply in conjunction with an early stopping criterion based on the squared-error loss. The extension of the CES method to the conformalized quantile regression (CQR) method of Romano et al. (2019) is developed explicitly in Appendix A7.

2.7. Implementation Details and Computational Cost

Beyond the cost of training the neural network (which is relatively expensive but does not need to be repeated for different test points) and the storage cost associated with saving the candidate models (which we have argued to be feasible in many applications), CES is quite computationally efficient. Firstly, CES treats all test points individually and could process them in parallel, although many operations do not need to be repeated. In particular, one can recycle the evaluation of the calibration loss across different test points; e.g., see (4). Thus, the model selection component can be easily implemented at cost $\mathcal{O}((n_{\text{es-cal}}+n_{\text{test}})\cdot T+n_{\text{test}}\cdot n_{\text{es-cal}})$ for classification (of which outlier detection is a special case) and $\mathcal{O}((n_{\text{es-cal}}+n_{\text{test}})\cdot T\cdot \log T+n_{\text{test}}\cdot n_{\text{es-cal}})$ for regression, where $n_{\text{es-cal}}=|\mathcal{D}_{\text{es-cal}}|$ and T is the number of candidate models. Note that the $T\cdot\log T$ dependence in the regression setting comes from the divide-and-conquer algorithm explained in Appendix A5.

It is possible that the cost of CES may become a barrier in some applications, particularly if T is very large, despite the slightly more than linear scaling. Hence, we recommend employing moderate values of T (e.g., 100 or 1000).

3. Numerical Experiments

3.1. Outlier Detection

The use of CES for outlier detection is demonstrated using the *CIFAR10* data set (Krizhevsky, 2009), a collection of 60,000 32-by-32 RGB images from 10 classes including common objects and animals. A convolutional neural network with ReLU activation functions is trained on a subset of the data to minimize the cross-entropy loss. The maximum number of epochs is set to be equal to 50. The trained classification model is then utilized to compute conformity scores for outlier detection with the convention that cats are inliers and the other classes are outliers. In particular, a nonconformity score for each Z_{n+1} is defined as 1 minus the output of the soft-max layer corresponding to the label "cat". This can be interpreted as an estimated probability of Z_{n+1} being an outlier. After translating these scores into a conformal p-value $\hat{u}_0(Z_{n+1})$, the null hypothesis that Z_{n+1} is a cat is rejected if $\hat{u}_0(Z_{n+1})\leq\alpha=0.1$.

The total number of samples utilized for training, early stopping, and conformal calibration is varied between 500 and 2000. In each case, CES is applied using 75% of the samples for training and 25% for early stopping and calibration. Note that the calibration step only utilizes inliers, while the other data subsets also contain outliers. The empirical performance of CES is measured in terms of the probability of falsely rejecting a true null hypothesis—the false positive rate (FPR)—and the probability of correctly rejecting a false null hypothesis—the true positive rate (TPR). The CES method is compared to three benchmarks. The first benchmark is naive early stopping with the best (*hybrid*) theoretical correction for the nominal coverage level described in Appendix A1.2. The second benchmark is early stopping based on data splitting, which utilizes 50% of the available samples for training, 25% for early stopping, and 25% for calibration. The third benchmark is full training without early stopping, which simply selects the model obtained after the last epoch. The test set consists of 100 independent test images, half of which are outliers. All results are averaged over 100 trials based on independent data subsets.

Figure 4 summarizes the performance of the four methods as a function of the total sample size; see Table A1 in Appendix A8 for the corresponding standard errors. All methods control the FPR below 10%, as expected, but CES achieves the highest TPR. The increased power of CES compared to data splitting is not surprising, as the latter relies on a less accurate model trained on less data. By contrast, the naive benchmark trains a model more similar to that of CES, but its TPR is not as high because the theoretical correction for the naive conformal p-values is overly pessimistic. Finally, full training is the least powerful competitor for large sample sizes because its underlying model becomes more and more overconfident as the training set grows.

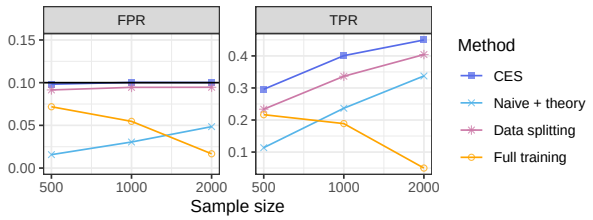


Figure 4. Average performance, as a function of the sample size, of conformal inferences for outlier detection based on neural networks trained and calibrated with different methods, on the *CIFAR10* data (Krizhevsky, 2009). Ideally, the TPR should be as large as possible while maintaining the FPR below 0.1. See Table A1 for more detailed results and standard errors.

Note that Table A1 also includes the results obtained with the naive benchmark detailed in Appendix A1, applied without the theoretical correction necessary to guarantee marginal coverage. Remarkably, the results show that the naive benchmark performs similarly to the CES method, even though only the latter has the advantage of enjoying rigorous finite-sample guarantees.

3.2. Multi-class Classification

The same *CIFAR10* data (Krizhevsky, 2009) are utilized here to demonstrate the performance of CES for a 10-class classification task. These experiments are conducted similarly to those in Section 3.1. The only difference is that now the soft-max output of the convolutional neural network is translated into conformal prediction sets, as explained in Appendix A4, instead of conformal p-values. The CES method is compared to the same three benchmarks adopted in Section 3.1. All prediction sets are calibrated to guarantee 90% marginal coverage, and their performances are evaluated based on cardinality.

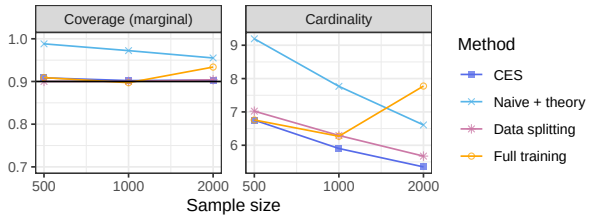


Figure 5. Average performance, as a function of the sample size, of conformal prediction sets for multi-class classification based on neural networks trained and calibrated with different methods, on the *CIFAR10* data (Krizhevsky, 2009). Ideally, the coverage should be close to 90% and the cardinality should be small. See Table A2 for more detailed results and standard errors.

Figure 5 summarizes the results averaged over 100 independent realizations of these experiments, while Table A2 in Appendix A8 reports on the corresponding standard errors. While all approaches always achieve the nominal coverage level, the CES method is able to do so with the smallest, and hence most informative, prediction sets. As before, the more disappointing performance of the data splitting benchmark can be explained by the more limited amount of data available for training, that of the naive benchmark by the excessive conservativeness of its theoretical correction, and that of the full training benchmark by overfitting. Note that Table A2 also includes the results obtained with the naive benchmark without the theoretical correction, which again performs similarly to CES.

3.3. Regression

We now apply the CES method to the following 3 public-domain regression data sets from the UCI Machine Learning repository (Pınar Tüfekci, 2012): physicochemical properties of protein tertiary structure (*bio*) (dat, b), hourly and daily counts of rental bikes (*bike*) (dat, a), and concrete compressive strength (*concrete*) (dat, c). These data sets were previously also considered by Romano et al. (2019), to which we refer for further details. As in the previous sections, we compare CES to the usual three benchmarks: naive early stopping with the *hybrid* theoretical correction for the nominal coverage level, early stopping based on data splitting, and full model training without early stopping. All methods utilize the same neural network with two hidden layers of width 128 and ReLU activation functions, trained for up to 1000 epochs. The models are calibrated in such a way as to produce conformal prediction sets with guaranteed 90% marginal coverage for a test set of 100 independent data points. The total sample size available for training, early stopping and calibration is varied between 200 and 2000. These data are allocated for specific training, early-stopping, and calibration operations as in Sections 3.1–3.2. The performance of each method is measured in terms of marginal coverage, worst-slab conditional coverage (Cauchois et al., 2021)—estimated as described in Sesia & Candès (2020)—and average width of the prediction intervals. All results are averaged over 100 independent experiments, each based on a different random sample from the original raw data sets.

Figure 2 summarizes the performance of the four alternative methods on the *bio* data, as a function of the total sample size; see Table A10 in Appendix A8 for the corresponding standard errors. These results show that all methods reach 90% marginal coverage in practice, as anticipated by the mathematical guarantees, although the theoretical correction for the naive early stopping method appears to be overly conservative. The CES method clearly performs best, in the sense that it leads to the shortest prediction intervals while also achieving approximately valid conditional coverage. By contrast, the conformal prediction intervals obtained without early stopping have significantly lower conditional coverage, which is consistent with the prior intuition that

fully trained neural networks can sometimes suffer from overfitting. More detailed results from these experiments can be found in Table A3 in Appendix A8. Analogous results corresponding to the *bike* and *concrete* data sets can be found in Figures A10–A11 and Tables A4–A5 in Appendix A8. Tables A3–A5 also include the results obtained with the naive benchmark applied without the necessary theoretical correction, which performs similarly to CES.

Finally, it must be noted that the widths of the prediction intervals output by the CES method in these experiments are very similar to those of the corresponding intervals produced by naively applying early stopping without data splitting and without the theoretical correction described in Appendix A1. This naive approach was not taken as a benchmark because it does not guarantee valid coverage, unlike the other methods. Nonetheless, it is interesting to note that the rigorous theoretical properties of the CES method do not come at the expense of a significant loss of power compared to this very aggressive heuristic, and in this sense, one may say that the conformal inferences computed by CES are "almost free".

3.4. Quantile Regression

Appendix A8 contains the results of additional numerical experiments investigating the performance of the CES quantile regression method mentioned in Section 2.6 and explained in Appendix A7. These experiments are carried out using the following publicly available and commonly investigated regression data sets from the UCI Machine Learning repository (Pınar Tüfekci, 2012): medical expenditure panel survey number 21 (*MEPS_21*) (mep); blog feedback (*blog_data*) (blo); Tennessee's student teacher achievement ratio (*STAR*) (Achilles et al., 2008); community and crimes (*community*) (com); physicochemical properties of protein tertiary structure (*bio*) (dat, b); house sales in King County (*homes*) (hom); and hourly and daily counts of rental bikes (*bike*) (dat, a). These data sets were previously also considered by Romano et al. (2019). The results demonstrate that the CES method compares favorably, in terms of the conditional coverage and average length of the output prediction intervals, to the analogous quantile regression versions of the benchmark approaches considered above.

4. Discussion

This paper has focused on early stopping and conformal calibration because these are two popular techniques, respectively designed to mitigate overfitting and reduce overconfidence, that were previously combined without much thought. However, the relevance of our methodology extends well beyond the problem considered in this paper. In fact, related ideas have already been utilized in the context of outlier detection to tune hyper-parameters and select the most promising candidate from an arbitrary toolbox of machine learning models (Liang et al., 2022). The techniques developed in this paper also allow one to calibrate, without further data splits, the most promising model selected in a data-driven way from an arbitrary machine learning toolbox in the context of multi-class classification and regression.

As detailed in Appendix A1, the naive benchmark that uses the same hold-out data twice, both for standard early stopping and standard conformal calibration, is not theoretically valid without conservative corrections. Nonetheless, our numerical experiments have shown that this naive approach often performs similarly to CES in practice. Of course, the naive benchmark may sometimes fail, and thus we would advise practitioners to apply the theoretically principled CES whenever its additional memory costs are not prohibitive. However, the empirical evidence suggests the naive benchmark may not be a completely unreasonable heuristic when CES is not applicable.

Software implementing the algorithms and data experiments are available online at https://github.com/ZiyiLiang/Conformalized_early_stopping.

Acknowledgements

The authors thank the Center for Advanced Research Computing at the University of Southern California for providing computing resources to carry out numerical experiments. The authors are also grateful to three anonymous reviewers for their insightful comments and suggestions. M. S. and Y. Z. are supported by NSF grant DMS 2210637. M. S. is also supported by an Amazon Research Award.

References

Blogfeedback dataset. https://github.com/xinbinhuang/feature-selection_blogfeedback. Accessed: Mar, 2023.

Communities and crime dataset. https://github.com/vbordalo/Communities-Crime. Accessed: Mar, 2023.

Bike sharing dataset. https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset, a. Accessed: July, 2019.

Physicochemical properties of protein tertiary structure data set. https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure, b. Accessed: July, 2019.

Concrete compressive strength data set. http://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength, c. Accessed: July, 2019.

House prices from king county dataset. https://www.kaggle.com/datasets/shivachandel/kc-house-data?select=kc_house_data.csv. Accessed: Mar, 2023.

Medical expenditure panel survey, panel 21. https://meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-192. Accessed: January, 2019.

Abramowitz, M. and Stegun, I. A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, ninth dover printing, tenth gpo printing edition, 1964.

Achilles, C., Bain, H. P., Bellott, F., Boyd-Zaharias, J., Finn, J., Folger, J., Johnston, J., and Word, E. Tennessee's Student Teacher Achievement Ratio (STAR) project, 2008. URL https://doi.org/10.7910/DVN/SIWH9F.

Angelopoulos, A. N., Bates, S., Jordan, M., and Malik, J. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. The limits of distribution-free conditional predictive inference. *Information and Inference*, 10(2):455–482, 2021a.

Barber, R. F., Candès, E. J., Ramdas, A., Tibshirani, R. J., et al. Predictive inference with the jackknife+. *Ann. Stat.*, 49(1):486–507, 2021b.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. Conformal prediction beyond exchangeability. *arXiv preprint arXiv:2202.13415*, 2022.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. Testing for outliers with conformal p-values. *Ann. Stat.*, 51(1):149 – 178, 2023.

Bellotti, A. Optimized conformal classification using gradient descent approximation. *arXiv preprint arXiv:2105.11255*, 2021.

Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. (B)*, 57(1):289–300, 1995.

Cauchois, M., Gupta, S., and Duchi, J. C. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *J. Mach. Learn. Res.*, 22: 81–1, 2021.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021.

Colombo, N. and Vovk, V. Training conformal predictors. In *Conformal and Probabilistic Prediction and Applications*, pp. 55–64. PMLR, 2020.

Devillers, O. and Golin, M. J. Incremental algorithms for finding the convex hulls of circles and the lower envelopes of parabolas. *Information Processing Letters*, 56(3):157–164, 1995.

Einbinder, B.-S., Romano, Y., Sesia, M., and Zhou, Y. Training uncertainty-aware classifiers with conformalized deep learning. In *Adv. Neural Inf. Process. Syst.*, volume 35, 2022.

Gibbs, I. and Candès, E. Conformal inference for online prediction with arbitrary distribution shifts. *arXiv preprint arXiv:2208.08401*, 2022.

Guan, L. and Tibshirani, R. Prediction and outlier detection in classification problems. *J. R. Stat. Soc. (B)*, 84(2): 524–546, 2022.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

Hechtlinger, Y., Póczos, B., and Wasserman, L. Cautious deep learning, 2018. arXiv:1805.09460.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

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

Kim, B., Xu, C., and Barber, R. Predictive inference is free with the jackknife+-after-bootstrap. *Advances in Neural Information Processing Systems*, 33:4138–4149, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *preprint at arXiv:1412.6980*, 2014.

Koenker, R. and Bassett, G. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

Krizhevsky, A. Learning multiple layers of features from tiny images, 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html.

Lei, J. and Wasserman, L. Distribution-free prediction bands for non-parametric regression. *J. R. Stat. Soc. (B)*, 76(1): 71–96, 2014.

Lei, J., Robins, J., and Wasserman, L. Distribution-free prediction sets. *J. Am. Stat. Assoc.*, 108(501):278–287, 2013.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. Distribution-free predictive inference for regression. *J. Am. Stat. Assoc.*, 113(523):1094–1111, 2018a.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. Distribution-free predictive inference for regression. *J. Am. Stat. Assoc.*, 113(523):1094–1111, 2018b.

Liang, Z., Sesia, M., and Sun, W. Integrative conformal p-values for powerful out-of-distribution testing with labeled outliers. *arXiv preprint arXiv:2208.11111*, 2022.

Marandon, A., Lei, L., Mary, D., and Roquain, E. Machine learning meets false discovery rate. *arXiv preprint arXiv:2208.06685*, 2022.

Marcus, G. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

Nielsen, F. and Yvinec, M. An output-sensitive convex hull algorithm for planar objects. *International Journal of Computational Geometry & Applications*, 8(01):39–65, 1998.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Prechelt, L. Automatic early stopping using cross validation: quantifying the criteria. *Neural networks*, 11(4):761–767, 1998.

Pınar Tüfekci, H. K. UCI machine learning repository, 2012. URL https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant.

Romano, Y., Patterson, E., and Candès, E. J. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, pp. 3538–3548, 2019.

Romano, Y., Barber, R. F., Sabatti, C., and Candès, E. With malice toward none: Assessing uncertainty via equalized coverage. *Harvard Data Science Review*, 2020a.

Romano, Y., Sesia, M., and Candès, E. J. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33, 2020b.

Salimans, T. and Kingma, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Saunders, C., Gammerman, A., and Vovk, V. Transduction with confidence and credibility. In *IJCAI*, 1999.

Sesia, M. and Candès, E. J. A comparison of some conformal quantile regression methods. *Stat*, 9(1), 2020.

Sesia, M. and Romano, Y. Conformal prediction using conditional histograms. *Advances in Neural Information Processing Systems*, 34, 2021.

Sesia, M., Favaro, S., and Dobriban, E. Conformal frequency estimation with sketched data under relaxed exchangeability. *arXiv preprint arXiv:2211.04612*, 2022.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Smith, J., Nouretdinov, I., Craddock, R., Offer, C., and Gammerman, A. Conformal anomaly detection of trajectories with a multi-class hierarchy. In *International symposium on statistical learning and data sciences*, pp. 281–290. Springer, 2015.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

Steinwart, I. and Christmann, A. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17 (1):211 – 225, 2011.

Stutz, D., Dvijotham, K., Cemgil, A. T., and Doucet, A. Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*, 2021.

Taylor, J. W. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. *Journal of Forecasting*, 19(4):299–311, 2000.

Thulasidasan, S., Chennupati, G., Bilmes, J. A., Bhattacharya, T., and Michalak, S. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Adv. Neural. Inf. Process. Syst.*, pp. 13888–13899, 2019.

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Vovk, V. Conditional validity of inductive conformal predictors. In *Proceedings of the Asian Conference on Machine Learning*, volume 25, pp. 475–490, 2012.

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

Vovk, V., Lindsay, D., Nouretdinov, I., and Gammerman, A. Mondrian confidence machine. Technical report, Royal Holloway, University of London, 2003. On-line Compression Modelling project.

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

Yang, Y. and Kuchibhotla, A. K. Finite-sample efficient conformal prediction. *arXiv preprint arXiv:2104.13871*, 2021.

A1. Naive Early Stopping Benchmarks

A1.1. Detailed Implementation of the Naive Benchmarks

We detail here the implementation of the naive benchmark considered in this paper, which consist of performing standard conformal calibration using the same hold-out samples utilized by standard early stopping techniques. This does not yield theoretically valid conformal inferences in finite samples because greedily utilizing the same hold-out data set twice, both to evaluate the early stopping criterion and to perform conformal calibration, will break the necessary exchangeability with the test point. Nonetheless, this approach can serve as an informative benchmark and it becomes useful in Appendix A6 to extend our rigorous conformalized early stopping method for regression problems in such as way as to explicitly avoid returning empty prediction intervals. For completeness, we present the implementation of the naive benchmark separately for outlier detection, multi-class classification, and regression, respectively in Algorithms A1, A2 and A3. Note that Algorithm A2 also allows for the possibility of computing prediction sets seeking (approximate) marginal coverage instead of (approximate) label-conditional coverage for multi-class classification problems; see Appendix A3 for further details on multi-class classification with marginal coverage.

Algorithm A1 Naive conformal outlier detection benchmark with greedy early stopping

Input: Exchangeable data points $Z_1, \ldots, Z_n$; test point Z_{n+1}.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
One-class classifier trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the one-class classifier for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
Pick the most promising model $t^* \in [T]$ minimizing $\mathcal{L}_{\text{es-cal}}(M_t)$ in (1), using the data in $\mathcal{D}_{\text{es-cal}}$.
Compute nonconformity scores $\hat{S}_i(Z_{n+1})$ for all $i \in \mathcal{D}_{\text{es-cal}} \cup \{n+1\}$ using model t^*.
Output: Naive conformal p-value $\hat{u}_0^{\text{naive}}(Z_{n+1})$ given by (3).

Algorithm A2 Naive conformal multi-class classification benchmark with greedy early stopping

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with labels $Y_i \in [K]$.
Test point with features X_{n+1}. Desired coverage level $1 - \alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
K-class classifier trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the K-class classifier for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
Pick the most promising model $t^* \in [T]$ minimizing $\mathcal{L}_{\text{es-cal}}(M_t)$ in (4), using the data in $\mathcal{D}_{\text{es-cal}}$.
for $y \in [K]$ **do**
  if Label-conditional coverage is desired **then**
    Define $\mathcal{D}^y_{\text{es-cal}} = \{i \in \mathcal{D}_{\text{es-cal}} : Y_i = y\}$.
    Compute scores $\hat{S}^y_i(X_{n+1})$ for all $i \in \mathcal{D}^y_{\text{es-cal}} \cup \{n+1\}$ using model t^*; see Appendix A4.
    Compute the naive conformal p-value $\hat{u}^{\text{naive}}_y(X_{n+1})$ according to (6).
  else
    Compute scores $\hat{S}^y_i(X_{n+1})$ for all $i \in \mathcal{D}_{\text{es-cal}} \cup \{n+1\}$ using model t^*; see Appendix A4.
    Compute the naive conformal p-value $\hat{u}^{\text{naive}}_y(X_{n+1})$ according to

$$\hat{u}^{\text{naive}}_y(X_{n+1}) = \frac{1 + |i \in \mathcal{D}_{\text{es-cal}} : \hat{S}^y_i(X_{n+1}) \leq \hat{S}^y_{n+1}(X_{n+1})|}{1 + |\mathcal{D}_{\text{es-cal}}|}.$$

  end if
end for
Output: Naive prediction set $\hat{C}^{\text{naive}}_\alpha(X_{n+1})$ given by (7).

Algorithm A3 Naive conformal regression benchmark with greedy early stopping

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with outcomes $Y_i \in \mathbb{R}$.
Test point with features X_{n+1}. Desired coverage level $1 - \alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
Regression model trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the regression model for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
Pick the most promising model $t^* \in [T]$ minimizing $\mathcal{L}_{\text{es-cal}}(M_t)$ in (8).
Evaluate nonconformity scores $\hat{S}_i(X_{n+1}) = |Y_i - \hat{\mu}_{t^*}(X_i)|$ for all $i \in \mathcal{D}_{\text{es-cal}}$.
Compute $\hat{Q}_{1-\alpha}(X_{n+1}) = \lceil (1-\alpha)(1 + |\mathcal{D}_{\text{es-cal}}|) \rceil$-th largest value in $\hat{S}_i(X_{n+1})$ for $i \in \mathcal{D}_{\text{es-cal}}$.
Output: Naive prediction interval $\hat{C}_\alpha^{\text{naive}}(X_{n+1}) = \hat{\mu}_{t^*}(X_{n+1}) \pm \hat{Q}_{1-\alpha}(X_{n+1})$.

A1.2. Theoretical Analysis of the Naive Benchmark

Although the naive benchmarks described above often perform similarly to CES in practice, they do not enjoy the same desirable theoretical guarantees. Nonetheless, we can study their behaviour in sufficient detail as to prove that their inferences are too far from being valid. Unfortunately, as demonstrated in Section 3, these theoretical results are still not tight enough to be very useful in practice. For simplicity, we will begin by focusing on outlier detection.

Review of existing results based on the DKW inequality. Yang & Kuchibhotla (2021) have recently studied the finite-sample coverage rate of a conformal prediction interval formed by naively calibrating a model selected among T possible candidates based on its performance on the calibration data set itself, which we denote by $\mathcal{D}_{\text{es-cal}}$. Although Yang & Kuchibhotla (2021) focus on conformal prediction intervals, here we find it easier to explain their ideas in the context of conformal p-values for outlier detection.

Let $\hat{S}_i(Z_{n+1}; t)$, for all $i \in \mathcal{D}_{\text{es-cal}}$ and $t \in [T]$, denote the nonconformity scores corresponding to model t, and denote the $\lfloor \alpha(1 + |\mathcal{D}_{\text{es-cal}}|) \rfloor$-th largest value in $\hat{S}_i(X_{n+1}; t)$ as $\hat{Q}_\alpha(Z_{n+1}; t)$. Let t^* indicate the selected model. As we are interested in constructing a conformal p-value $\hat{u}_0^{\text{naive}}(Z_{n+1})$, the goal is to bound from above the tail probability

$$\mathbb{P}\left(\hat{u}_0^{\text{naive}}(Z_{n+1}) > \alpha\right) = \mathbb{E}\left[\mathbb{P}\left(\hat{S}_i(X_{n+1}; t^*) > \hat{Q}_\alpha(Z_{n+1}; t^*) \mid \mathcal{D}_{\text{es-cal}}\right)\right]. \tag{A14}$$

Intuitively, if $n_{\text{es-cal}} = |\mathcal{D}_{\text{es-cal}}|$ is sufficiently large, the conditional probability inside the expected value on the right-hand-side above can be well-approximated by the following empirical quantity:

$$\frac{1}{n} \sum_{i \in \mathcal{D}_{\text{es-cal}}} \mathbb{1}\left\{\hat{S}_i(X_{n+1}; t^*) > \hat{Q}_\alpha(Z_{n+1}; t^*)\right\} = \frac{\lceil (1 + n_{\text{es-cal}})(1-\alpha) \rceil}{n_{\text{es-cal}}} \geq \left(1 + \frac{1}{n_{\text{es-cal}}}\right)(1-\alpha).$$

The quality of this approximation in finite samples can be bound by the DKW inequality, which holds for any $\varepsilon \geq 0$:

$$\mathbb{P}\left(\sup_{s \in \mathbb{R}} \left| \frac{1}{n_{\text{es-cal}}} \sum_{i \in \mathcal{D}_{\text{es-cal}}} \mathbb{1}\left\{\hat{S}_i(X_{n+1}; t^*) > s\right\} - \mathbb{P}\left(\hat{S}_i(X_{n+1}; t^*) > s \mid \mathcal{D}_{\text{es-cal}}\right) \right| > \varepsilon \right) \leq 2e^{-2n_{\text{es-cal}}\varepsilon^2}. \tag{A15}$$

Starting from this, Theorem 1 in Yang & Kuchibhotla (2021) shows that

$$\mathbb{P}(\hat{u}_0^{\text{naive}}(Z_{n+1}) > \alpha) \geq \left(1 + \frac{1}{n_{\text{es-cal}}}\right)(1-\alpha) - \frac{\sqrt{\log(2T)/2} + c(T)}{\sqrt{n_{\text{es-cal}}}}, \tag{A16}$$

where $c(T)$ is a constant that can be computed explicitly and is generally smaller than $1/3$. Intuitively, the $[\sqrt{\log(2T)/2} + c(T)]/\sqrt{n_{\text{es-cal}}}$ term above can be interpreted as the worst-case approximation error among all possible models $t \in [T]$.

One limitation with this result is that is gives a worst-case correction that does not depend on the chosen level α, and one would intuitively expect this bound to be tighter for $\alpha = 1/2$ and overly conservative for the small α values (e.g., $\alpha = 0.1$) that are typically interesting in practice. (This intuition will be confirmed empirically in Figure A7.) This observation motivates the following alternative analysis, which can often give tighter results.

Alternative probabilistic bound based on Markov's inequality. Define $W_t = \mathbb{P}\left[\hat{u}_0^{\text{naive}}(Z_{n+1}; t) > \alpha \mid \mathcal{D}_{\text{es-cal}}\right]$. Lemma 3 in Vovk (2012) tells us that W_t follows a Beta distribution, assuming exchangeability among $\mathcal{D}_{\text{es-cal}}$ and the test point. That is,

$$W_t \sim \text{Beta}(n_{\text{es-cal}} + 1 - l, l), \quad l = \lfloor \alpha(n_{\text{es-cal}} + 1) \rfloor.$$

In the following, we will denote the corresponding inverse Beta cumulative distribution function as $I^{-1}(x; n_{\text{es-cal}} + 1 - l, l)$. This result can be used to derive an alternative upper bound for $\mathbb{P}(\hat{u}_0^{\text{naive}}(Z_{n+1}) > \alpha)$ using the Markov's inequality.

Proposition A1. *Assume $Z_1, \ldots, Z_n, Z_{n+1}$ are exchangeable random samples, and let $\hat{u}_0^{\text{naive}}(Z_{n+1})$ be the output of Algorithm A1, for any given $\alpha \in (0, 1)$. Then, for any fixed $\alpha \in (0, 1)$ and any $b > 1$, letting $l = \lfloor \alpha(n_{\text{es-cal}} + 1) \rfloor$,*

$$\mathbb{P}\left[\hat{u}_0^{\text{naive}}(Z_{n+1}) > \alpha\right] \geq I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} + 1 - l, l\right) \cdot (1 - 1/b).$$

Note that this bound depends on α in a more complex way compared to that of Yang & Kuchibhotla (2021). However, its asymptotic behaviour in the large-T limit remains similar, as shown below.

Lemma A2. *Denote $I^{-1}(x; n_{\text{es-cal}} + 1 - l, l)$ as the inverse Beta cumulative distribution function. For any fixed $b > 1$ and $\alpha \in (0, 1)$, letting $l = \lfloor \alpha(n_{\text{es-cal}} + 1) \rfloor$, for sufficiently large T and $n_{\text{es-cal}}$, we have:*

$$I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} + 1 - l, l\right) = (1 - \alpha) - \sqrt{\frac{\alpha(1-\alpha)}{n_{\text{es-cal}} + 1}} \cdot \sqrt{2\log(bT)} + O\left(\frac{1}{\sqrt{n_{\text{es-cal}} \log(T)}}\right).$$

In simpler terms, Lemma A2 implies that the coverage lower bound in Proposition A1 is approximately equal to

$$\left[(1 - \alpha) - \sqrt{\frac{\alpha(1-\alpha)}{n_{\text{es-cal}} + 1}} \cdot \sqrt{2\log(bT)}\right] \cdot \left(1 - \frac{1}{b}\right),$$

which displays an asymptotic behaviour similar to that of the bound from Yang & Kuchibhotla (2021). Further, the Markov bound is easy to compute numerically and often turns out to be tighter as long as b is moderately large (e.g., $b = 100$), as we shall see below. Naturally, the same idea can also be applied to bound the coverage of naive conformal prediction sets or intervals output by Algorithm A2 or Algorithm A3, respectively.

Corollary A3. *Assume $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, Y_{n+1})$ are exchangeable random sample, and let $\hat{C}_\alpha^{\text{naive}}(X_{n+1})$ be the output of Algorithm A2, for any given $\alpha \in (0, 1)$. Then, for any $b > 1$, letting $l = \lfloor \alpha(n_{\text{es-cal}} + 1) \rfloor$,*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha^{\text{naive}}(X_{n+1})\right] \geq I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} + 1 - l, l\right) \cdot (1 - 1/b).$$

Corollary A4. *Assume $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, Y_{n+1})$ are exchangeable random samples, and let $\hat{C}_\alpha^{\text{naive}}(X_{n+1})$ be the output of Algorithm A3, for any $\alpha \in (0, 1)$. Then, for any $b > 1$, letting $l = \lfloor \alpha(n_{\text{es-cal}} + 1) \rfloor$,*

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha^{\text{naive}}(X_{n+1})\right] \geq I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} + 1 - l, l\right) \cdot (1 - 1/b).$$

Hybrid probabilistic bound. Since neither the DKW nor the Markov bound described above always dominate the other for all possible combinations of T, $n_{\text{es-cal}}$, and α, it makes sense to combine them to obtain a uniformly tighter *hybrid* bound. For any fixed $b > 1$ and any T, $n_{\text{es-cal}}$, and α, define $H(T, n_{\text{es-cal}}, \alpha)$ as

$$H(T, n_{\text{es-cal}}, \alpha) = \max\left\{I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} + 1 - l, l\right) \cdot (1 - 1/b), \left(1 + \frac{1}{n_{\text{es-cal}}}\right)(1 - \alpha) - \frac{\sqrt{\log(2T)/2} + c(T)}{\sqrt{n_{\text{es-cal}}}}\right\}.$$

It then follows immediately from Yang & Kuchibhotla (2021) and Proposition A1 that, under the same conditions of

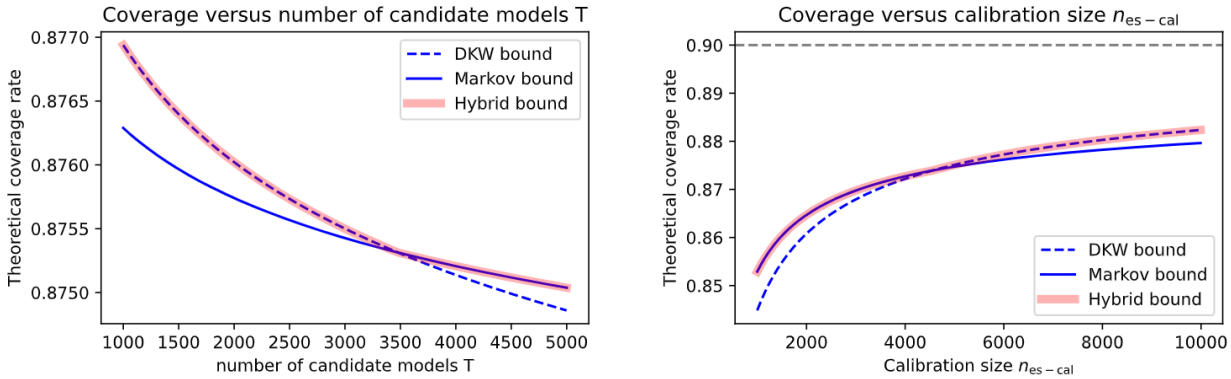


Figure A6. Numerical comparison of different theoretical lower bounds for the marginal coverage of conformal prediction sets computed with a naive early stopping benchmark (e.g., Algorithm A2). Left: lower bounds for the marginal coverage as a function of the number of candidate models T, when $\alpha = 0.1$ and $n_{\text{es-cal}} = 8000$. Right: lower bounds for the marginal coverage as a function of the number of hold-out data points, $n_{\text{es-cal}}$, when $\alpha = 0.1$ and $T = 100$. Higher values correspond to tighter bounds.

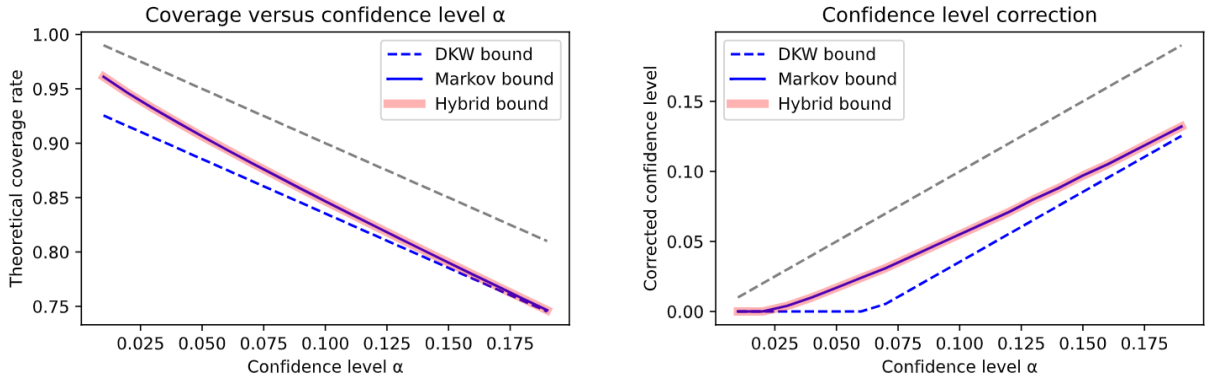


Figure A7. Numerical comparison of different theoretical lower bounds for the marginal coverage of conformal prediction sets computed with a naive early stopping benchmark (e.g., Algorithm A2), as a function of the nominal significance level α. Left: lower bounds for the marginal coverage as a function of α, when $T = 1000$ and $n_{\text{es-cal}} = 1000$. Right: theoretically corrected significance level necessary needed to achieve the marginal coverage guarantees expected at the nominal α level, as a function of α when $T = 1000$ and $n_{\text{es-cal}} = 1000$. The dashed grey lines indicate the ideal values corresponding to standard conformal inferences based on calibration data that are independent of those used for early stopping. Higher values correspond to tighter bounds.

Proposition A1, for any fixed $b > 1$,

$$\mathbb{P}\left[\hat{u}_0^{\text{naive}}(Z_{n+1}) > \alpha\right] \geq H(T, n_{\text{es-cal}}, \alpha).$$

Of course, the same argument can also be utilized to tighten the results of Corollaries A3–A4.

Numerical comparison of different probabilistic bounds. Figure A6 compares the three probabilistic bounds described above (*DKW*, *Markov*, and *hybrid*) as a function of the number of candidate models T and of the number of hold-out data points $n_{\text{es-cal}}$, in the case of $\alpha = 0.1$. For simplicity, the Markov and hybrid bounds are evaluated by setting $b = 100$, which may not be the optimal choice but appears to work reasonably well. These results show that Markov bound tends to be tighter than the DKW bound for large values of T and for small values of $n_{\text{es-cal}}$, while the hybrid bound generally achieves the best of both worlds. Lastly, Figure A7 demonstrates that the Markov bound tends to be tighter when α is small. The Markov and hybrid bounds here are also evaluated using $b = 100$.

A2. Main Algorithms

Algorithm A4 Conformalized early stopping for outlier detection

Input: Exchangeable data points $Z_1, \ldots, Z_n$; test point Z_{n+1}.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
One-class classifier trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the one-class classifier for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
Pick the most promising model $\hat{M}_{\text{ces}}(Z_{n+1})$ according to (2), using the data in $\mathcal{D}_{\text{es-cal}} \cup \{n+1\}$.
Compute nonconformity scores $\hat{S}_i(Z_{n+1})$ for all $i \in \mathcal{D}_{\text{es-cal}} \cup \{n+1\}$ using model $\hat{M}_{\text{ces}}(Z_{n+1})$.
Output: Conformal p-value $\hat{u}_0(Z_{n+1})$ given by (3).

Algorithm A5 Conformalized early stopping for multi-class classification

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with labels $Y_i \in [K]$.
Test point with features X_{n+1}. Desired coverage level $1-\alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
K-class classifier trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the K-class classifier for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
for $y \in [K]$ **do**
  Define $\mathcal{D}^y_{\text{es-cal}} = \{i \in \mathcal{D}_{\text{es-cal}} : Y_i = y\}$ and imagine $Y_{n+1} = y$.
  Pick the model $\hat{M}_{\text{ces}}(X_{n+1}, y)$ according to (5), using the data in $\mathcal{D}_{\text{es-cal}} \cup \{n+1\}$.
  Compute scores $\hat{S}^y_i(X_{n+1})$ for all $i \in \mathcal{D}^y_{\text{es-cal}} \cup \{n+1\}$ using $\hat{M}_{\text{ces}}(X_{n+1}, y)$; see Appendix A4.
  Compute the conformal p-value $\hat{u}_y(X_{n+1})$ according to (6).
end for
Output: Prediction set $\hat{C}_\alpha(X_{n+1})$ given by (7).

Algorithm A6 Conformalized early stopping for regression

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with outcomes $Y_i \in \mathbb{R}$.
Test point with features X_{n+1}. Desired coverage level $1-\alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
Regression model trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the regression model for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
Evaluate $\hat{M}_{\text{ces}}(X_{n+1}, y)$ as in (10), using Algorithm A8.
Partition the domain of Y into L intervals $\mathcal{B}_l$, for $l \in [L]$, based on the knots of $\hat{M}_{\text{ces}}(X_{n+1}, y)$.
for $l \in [L]$ **do**
  Evaluate nonconformity scores $\hat{S}_i(X_{n+1}, \mathcal{B}_l)$ for all $i \in \mathcal{D}_{\text{es-cal}}$ as in (11).
  Compute $\hat{Q}_{1-\alpha}(X_{n+1}, \mathcal{B}_l)$ as the $\lceil (1-\alpha)(1+|\mathcal{D}_{\text{es-cal}}|) \rceil$-th largest value among $\hat{S}_i(X_{n+1}, \mathcal{B}_l)$.
  Construct the interval $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)$ according to (12).
end for
Output: Prediction interval $\hat{C}_\alpha(X_{n+1})$ given as a function of $\{\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)\}_{l=1}^L$ by (13).

A3. Classification with Marginal Coverage

The conformalized early stopping method presented in Section 2.4 can be easily modified to produce prediction sets with marginal rather than label-conditional coverage, as outlined in Algorithm A7. The difference between Algorithm A5 and Algorithm A7 is that the latter utilizes all calibration data in $\mathcal{D}_{\text{es-cal}}$ to compute each conformal p-value $\hat{u}_y(X_{n+1})$, not only the samples with true label y. An advantage of this approach is that conformal p-values based on a larger calibration samples are less aleatoric (Bates et al., 2023) and require less conservative finite-sample corrections (i.e., the "+1" term the numerator

of the p-value formula becomes more negligible as the calibration set size increases). In turn, this tends to lead to smaller prediction sets with potentially more stable coverage conditional on the calibration data (Sesia & Candès, 2020; Bates et al., 2023) Of course, the downside of these prediction sets is that they can only be guaranteed to provide marginal coverage, although they can sometimes also perform well empirically in terms of label-conditional coverage (Romano et al., 2020b).

Theorem A1. *Assume $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, Y_{n+1})$ are exchangeable random samples, and let $\hat{C}_\alpha(X_{n+1})$ be the output of Algorithm A7, for any $\alpha \in (0, 1)$. Then, $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})] \geq 1 - \alpha$.*

Algorithm A7 Conformalized early stopping for multi-class classification with marginal coverage

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with labels $Y_i \in [K]$.
Test point with features X_{n+1}. Desired coverage level $1 - \alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
K-class classifier trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the K-class classifier for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
for $y \in [K]$ **do**
  Imagine $Y_{n+1} = y$.
  Pick the model $\hat{M}_{\text{ces}}(X_{n+1}, y)$ according to (5), using the data in $\mathcal{D}_{\text{es-cal}} \cup \{n + 1\}$.
  Compute scores $\hat{S}_i(X_{n+1}, y)$ for all $i \in \mathcal{D}_{\text{es-cal}} \cup \{n + 1\}$ using $\hat{M}_{\text{ces}}(X_{n+1}, y)$; see Appendix A4.
  Compute the conformal p-value $\hat{u}_y^{\text{marg}}(X_{n+1})$ according to

$$\hat{u}_y^{\text{marg}}(X_{n+1}) = \frac{1 + |i \in \mathcal{D}_{\text{es-cal}} : \hat{S}_i^y(X_{n+1}) \leq \hat{S}_{n+1}^y(X_{n+1})|}{1 + |\mathcal{D}_{\text{es-cal}}|}. \tag{A17}$$

end for
Output: Prediction set $\hat{C}_\alpha(X_{n+1})$ given by (7), with $\hat{u}_y^{\text{marg}}(X_{n+1})$ instead of $\hat{u}_y(X_{n+1})$.

A4. Review of Nonconformity Scores for Classification

This section reviews the relevant background on the adaptive nonconformity scores for classification developed by Romano et al. (2020b). For any $x \in \mathcal{X}$ and $y \in [K]$, let $\hat{\pi}_y(x)$ denote any (possibly very inaccurate) estimate of the true $\mathbb{P}[Y = y \mid X = x]$ corresponding to the unknown data-generating distribution. Concretely, a typical choice of $\hat{\pi}$ may be given by the output of the final softmax layer of a neural network classifier, for example. For any $x \in \mathcal{X}$ and $\tau \in [0, 1]$, define the *generalized conditional quantile* function L, with input $x, \hat{\pi}, \tau$, as:

$$L(x; \hat{\pi}, \tau) = \min\{k \in [K] \ : \ \hat{\pi}_{(1)}(x) + \hat{\pi}_{(2)}(x) + \ldots + \hat{\pi}_{(k)}(x) \geq \tau\}, \tag{A18}$$

where $\hat{\pi}_{(1)}(x) \leq \hat{\pi}_{(2)}(x) \leq \ldots \hat{\pi}_{(K)}(x)$ are the order statistics of $\hat{\pi}_1(x) \leq \hat{\pi}_2(x) \leq \ldots \hat{\pi}_K(x)$. Intuitively, $L(x; \hat{\pi}, \tau)$ gives the size of the smallest possible subset of labels whose cumulative probability mass according to $\hat{\pi}$ is at least τ. Define also a function $\mathcal{S}$ with input x, $u \in (0, 1)$, $\hat{\pi}$, and τ that computes the set of most likely labels up to (but possibly excluding) the one identified by $L(x; \hat{\pi}, \tau)$:

$$\mathcal{S}(x, u; \hat{\pi}, \tau) = \begin{cases} \text{‘}y\text{’ indices of the } L(x; \hat{\pi}, \tau) - 1 \text{ largest } \hat{\pi}_y(x), & \text{if } u \leq V(x; \hat{\pi}, \tau), \\ \text{‘}y\text{’ indices of the } L(x; \hat{\pi}, \tau) \text{ largest } \hat{\pi}_y(x), & \text{otherwise,} \end{cases} \tag{A19}$$

where

$$V(x; \hat{\pi}, \tau) = \frac{1}{\hat{\pi}_{(L(x;\hat{\pi},\tau))}(x)} \left[\sum_{k=1}^{L(x;\hat{\pi},\tau)} \hat{\pi}_{(k)}(x) - \tau \right].$$

Then, define the *generalized inverse quantile* nonconformity score function s, with input $x, y, u; \hat{\pi}$, as:

$$s(x, y, u; \hat{\pi}) = \min\left\{\tau \in [0, 1] : y \in \mathcal{S}(x, u; \hat{\pi}, \tau)\right\}. \tag{A20}$$

Intuitively, $s(x, y, u; \hat{\pi})$ is the smallest value of τ for which the set $\mathcal{S}(x, u; \hat{\pi}, \tau)$ contains the label y. Finally, the nonconformity score for a data point (X_i, Y_i) is given by:

$$\hat{S}_i = s(X_i, Y_i, U_i; \hat{\pi}), \tag{A21}$$

where U_i is a uniform random variable independent of anything else. Note that this can also be equivalently written more explicitly as:

$$\hat{S}_i = \hat{\pi}_{(1)}(X_i) + \hat{\pi}_{(2)}(X_i) + \ldots + \hat{\pi}_{(r(Y_i, \hat{\pi}(X_i)))}(X_i) - U_i \cdot \hat{\pi}_{(r(Y_i, \hat{\pi}(X_i)))}(X_i), \tag{A22}$$

where $r(Y_i, \hat{\pi}(X_i))$ is the rank of Y_i among the possible labels $y \in [K]$ based on $\hat{\pi}_y(X_i)$, so that $r(y, \hat{\pi}(X_i)) = 1$ if $\hat{\pi}_y(X_i) = \hat{\pi}_{(1)}(X_i)$. The idea motivating this construction is that the nonconformity score $\hat{S}_i$ defined above is guaranteed to be uniformly distributed on $[0, 1]$ conditional on X if the model $\hat{\pi}$ estimates the true unknown $\mathbb{P}[Y = y \mid X = x]$ accurately for all $x \in \mathcal{X}$. This is a desirable property in conformal inference because it leads to statistically efficient prediction sets that can often achieve relatively high feature-conditional coverage in practice, even if the true data-generating distribution is such that some observations are much noisier than others; see Romano et al. (2020b) for further details.

Finally, we conclude this appendix by noting that the nonconformity scores in Section 2.4 are written as $\hat{S}_i(X_{n+1}, y)$, instead of the more compact notation $\hat{S}_i$ adopted here, simply to emphasize that they are computed based on class probabilities $\hat{\pi}$ estimated by a data-driven model $\hat{M}$ that depends on the test features X_{n+1} as well as on the placeholder label y for Y_{n+1}.

A5. Efficient Computation of the Lower Envelope

This section explains how to implement a computationally efficient divide-and-conquer algorithm for finding the lower envelope of a family of T parabolas or a family of shifted pinball loss functions at cost $\mathcal{O}(T \log T)$ (Devillers & Golin, 1995; Nielsen & Yvinec, 1998). This solution, outlined in Algorithm A8 and Algorithm A9, is useful to implement the proposed CES method for regression problems, as detailed in Algorithm A6 and Algorithm A11.

Algorithm A8 Divide-and-conquer algorithm for finding the lower envelope of many parabolas

1: **Input**: A set of parabolas $L = \{l_1, l_2, \ldots, l_T\}$ of forms $l_i = a_i x^2 + b_i x + c_i$ for $i = 1, \ldots, T$.
2: Randomly split L into two subsets. Repeat splitting until each subset only contains one parabola or is empty.
3: For each subset with only one parabola, set the parabola itself as the lower envelope and set the initial breakpoint list to $[-\infty, +\infty]$.
4: **for** each interval constructed by adjacent breakpoints **do**
5: Within the interval, identify the two parabolas contributing to the previous lower envelopes, denoted as P_1, P_2.
6: Evaluate P_1 and P_2 at the current interval endpoints.
7: Calculate the intersection point p of P_1 and P_2. There exists at most one such p because $a_i = 1, \forall i$, by (8).
8: **if** p not exists or p exists but lies outside the current interval **then**
9: Set the new lower envelope as the parabola with smaller values computed at the interval endpoints.
10: **else**
11: Add p as a breakpoint.
12: Within the current interval, set the new lower envelope below and above p based on evaluations of the parabolas at the interval endpoints.
13: **end if**
14: Update and sort the breakpoint list and update the new lower envelope.
15: **end for**
16: Recursively merge two lower envelopes to form a new lower envelope by repeating Lines 4–15.
17: **Output**: A sorted dictionary of breakpoints and parabola indices characterizing the lower envelope of L.

Algorithm A9 Divide-and-conquer algorithm for finding the lower envelope of many pinball loss functions

1: **Input**: A set of shifted pinball loss functions $L = \{l_1, l_2, \ldots, l_T\}$ of forms $l_i = c_i + \rho_\beta(y, \hat{y})$ for $i = 1, \ldots, T$.
2: Randomly split L into two subsets. Repeat splitting until each subset only contains one pinball loss function or is empty.
3: For each subset with only one pinball loss function, set the function itself as the lower envelope and set the initial breakpoint list to $[-\infty, +\infty]$.
4: **for** each interval constructed by adjacent breakpoints **do**
5: Within the interval, identify the two pinball loss functions contributing to the previous lower envelopes; i.e., P_1, P_2.
6: Evaluate P_1 and P_2 at the current interval endpoints.
7: Calculate the intersection point p of P_1 and P_2. There exists at most one such p because β is the same $\forall i$, by (A29).
8: **if** p not exists or p exists but lies outside the current interval **then**
9: Set the new lower envelope as the pinball loss function with smaller values computed at the interval endpoints.
10: **else**
11: Add p as a breakpoint.
12: Within the current interval, set the new lower envelope below and above p based on evaluations of the pinball loss functions at the interval endpoints.
13: **end if**
14: Update and sort the breakpoint list and update the new lower envelope.
15: **end for**
16: Recursively merge two lower envelopes to form a new lower envelope by repeating Lines 4–15.
17: **Output**: A sorted dictionary of breakpoints and pinball loss function indices characterizing the lower envelope of L.

A6. Avoiding Empty Predictions in CES for Regression

This section presents Algorithm A10, which extends Algorithm A6 from Section 2.5 in such a way as to explicitly avoid returning empty prediction intervals.

Algorithm A10 Conformalized early stopping for regression, avoiding empty predictions

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with outcomes $Y_i \in \mathbb{R}$.
Test point with features X_{n+1}. Desired coverage level $1 - \alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
Regression model trainable via (stochastic) gradient descent.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train the regression model for t^{max} epochs and save the intermediate models $M_{t_1}, \ldots, M_{t_T}$.
Evaluate $\hat{C}_\alpha(X_{n+1})$ using Algorithm A6.
if $\hat{C}_\alpha(X_{n+1}) = \emptyset$ **then**
Evaluate $\hat{C}_\alpha^{\text{naive}}(X_{n+1})$ using Algorithm A3. Set $\hat{C}_\alpha(X_{n+1}) = \hat{C}_\alpha^{\text{naive}}(X_{n+1})$.
end if
Output: A non-empty prediction interval $\hat{C}_\alpha(X_{n+1})$.

Corollary A1. *Assume $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, Y_{n+1})$ are exchangeable random samples, and let $\hat{C}_\alpha(X_{n+1})$ be the output of Algorithm A10, for any $\alpha \in (0, 1)$. Then, $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})] \geq 1 - \alpha$.*

A7. CES for Quantile Regression

A limitation of the CES method for regression described in Section 2.5 is that it is not adaptive to heteroscedasticity because it produces prediction intervals with constant length regardless of the input features (Romano et al., 2019). However, this limitation can be easily addressed by extending CES to the conformalized quantile regression (CQR) method of Romano et al. (2019), which is briefly reviewed below.

A7.1. Review of Conformalized Quantile Regression

This section reviews the relevant background on conditional quantile regression (Koenker & Bassett, 1978) and conformalized quantile regression (CQR) (Romano et al., 2019). In contrast to the classical regression models that estimate the conditional

mean of the test response Y_{n+1} given the test feature $X_{n+1} = x$, the quantile regression estimates the conditional quantile q_β of Y_{n+1} given $X_{n+1} = x$ defined as

$$q_\beta(x) = \inf\{y \in \mathbb{R} : \mathbb{P}(Y_{n+1} \leq y | X_{n+1} = x) \geq \beta\} \tag{A23}$$

This can be formulated as solving the optimization problem:

$$\hat{q}_\beta(x) = f(x, \hat{\theta}), \quad \hat{\theta} = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \rho_\beta(Y_i, f(X_i, \theta)) \tag{A24}$$

where $f(x, \theta)$ represents the quantile regression function (Koenker & Bassett, 1978) and ρ_β is the convex "pinball loss" function (Steinwart & Christmann, 2011), geometrically illustrated in Figure A8 and mathematically defined by

$$\rho_\beta(y, \hat{y}) = \begin{cases} \beta(y - \hat{y}) & \text{if } y - \hat{y} > 0, \\ (1 - \beta)(\hat{y} - y) & \text{otherwise} \end{cases} \tag{A25}$$

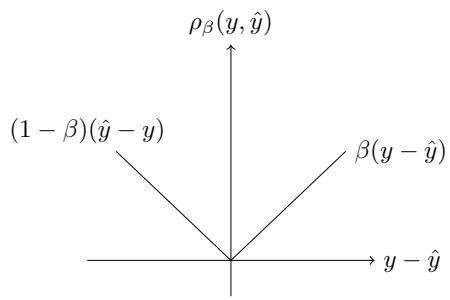


Figure A8. Visualization of the pinball loss function in (A25).

To construct an efficient prediction interval $\hat{C}(X_{n+1})$ whose length is adaptive to the local variability of X_{n+1} and satisfies the marginal coverage validity, i.e. $\mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1})) \geq 1 - \alpha$ for any $\alpha \in (0, 1)$, Yaniv Romano, Evan Patterson, and Emmanuel J. Candès developed CQR that combines conformal inference (Vovk et al., 1999) (Vovk et al., 2005) and conditional quantile regression (Koenker & Bassett, 1978). As in split conformal prediction, firstly the available data is randomly split into a proper training set, indexed by $\mathcal{I}_1$, and a calibration set, indexed by $\mathcal{I}_2$. Given any quantile regression algorithm $\mathcal{A}$, two conditional quantile functions $\hat{q}_{\alpha_{\text{lo}}}, \hat{q}_{\alpha_{\text{hi}}}$ are fitted on $\mathcal{I}_1$, where $\alpha_{\text{lo}} = \alpha/2$ and $\alpha_{\text{hi}} = 1 - \alpha/2$:

$$\{\hat{q}_{\alpha_{\text{lo}}}, \hat{q}_{\alpha_{\text{hi}}}\} \leftarrow \mathcal{A}(\{(X_i, Y_i) : i \in \mathcal{I}_1\}) \tag{A26}$$

In the next step, we compute the conformity scores on the calibration data set $\mathcal{I}_2$ as

$$E_i = \max\{\hat{q}_{\alpha_{\text{lo}}}(X_i) - Y_i, Y_i - \hat{q}_{\alpha_{\text{hi}}}(X_i)\} \quad \text{for } i \in \mathcal{I}_2 \tag{A27}$$

The conformity score designed in Equation A27 accounts for both undercoverage and overcoverage in the following sense: If Y_i is below the lower endpoint of the interval or above the upper endpoint of the interval, then E_i is the magnitude of the undercoverage error. Similarly, if Y_i correctly belongs to the interval, then E_i is non-positive. Finally, CQR constructs the prediction interval for the test response value Y_{n+1} as

$$\hat{C}(X_{n+1}) = [\hat{q}_{\alpha_{\text{lo}}}(X_{n+1}) - Q_{1-\alpha}(E, \mathcal{I}_2), \hat{q}_{\alpha_{\text{hi}}}(X_{n+1}) + Q_{1-\alpha}(E, \mathcal{I}_2)] \tag{A28}$$

where $Q_{1-\alpha}(E, \mathcal{I}_2)$ is the $(1 - \alpha)(1 + \frac{1}{|\mathcal{I}_2|})$-th empirical quantile of $\{E_i : i \in \mathcal{I}_2\}$

A7.2. CES for Quantile Regression

As in the previous section, consider a data set containing n exchangeable observations (X_i, Y_i), for $i \in \mathcal{D} = [n]$, and a test point (X_{n+1}, Y_{n+1}) with a latent label $Y_{n+1} \in \mathbb{R}$. First, the data in $\mathcal{D}$ are randomly split into two subsets, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$. The data in $\mathcal{D}_{\text{train}}$ are utilized to train a neural network quantile regression model (Taylor, 2000) by seeking to minimize the pinball loss instead of the squared error loss, for each target level $\beta = \beta_{\text{low}}$ and $\beta = \beta_{\text{high}}$ (e.g., $\beta_{\text{low}} = \alpha/2$ and $\beta_{\text{high}} = 1 - \alpha/2$). Note that the same neural network, with two separate output nodes, can be utilized to estimate conditional quantiles at two different levels; e.g., as in Romano et al. (2019). For any $t \in [t^{\max}]$, let $M_{\beta,t}$ denote the intermediate neural network model stored after t epochs of stochastic gradient descent, following the same notation as in Section 2.5. For each target level β and any $x \in \mathcal{X}$, let $\hat{q}_{\beta,t}(x)$ denote the approximate $\beta-$th conditional quantile of the unknown conditional distribution of $Y \mid X = x$ estimated by $M_{\beta,t}$.

Similarly to Section 2.5, for any model $M_{\beta,t}$ and any $x \in \mathcal{X}$, $y \in \mathbb{R}$, define the augmented loss evaluated on the calibration data including also a dummy test point (x, y) as:

$$\begin{aligned}\mathcal{L}^{+1}_{\text{es-cal}}(M_{\beta,t}, x, y) &= \mathcal{L}_{\text{es-cal}}(M_{\beta,t}) + \mathcal{L}(M_{\beta,t}, x, y) \\ &= \sum_{i \in \mathcal{D}_{\text{es-cal}}} \rho_\beta(Y_i, \hat{q}_{\beta,t}(X_i)) + \rho_\beta(y, \hat{q}_{\beta,t}(x)),\end{aligned} \tag{A29}$$

where ρ_β denotes the pinball loss function defined in (A25). For any model $M_{\beta,t}$, the augmented loss is equal to a constant plus a convex function of y, namely $\rho_\beta(y, \hat{q}_{\beta,t}(x))$. Therefore, for any fixed x, the quantity in (A29) can be sketched as a function of $M_{\beta,t}$ and y as shown in Figure A9. This is analogous to Figure 3 from Section 2.5, with the difference that now the quadratic functions have been replaced by piece-wise linear "pinball" functions.

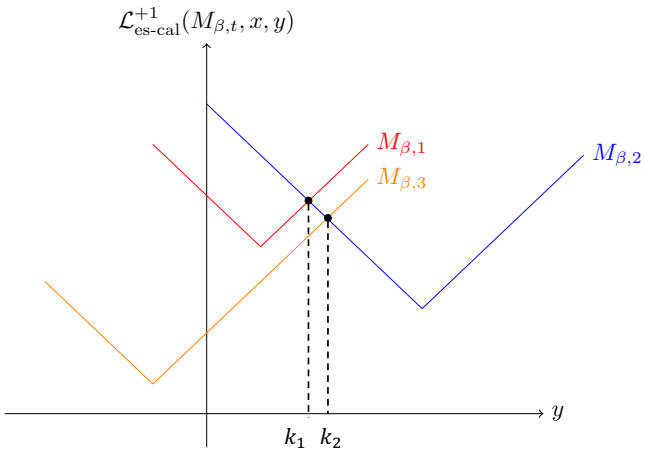


Figure A9. Pinball loss functions on test-augmented hold-out data for three alternative regression models, M_1, M_2 and M_3, as a function of the place-holder outcome y for the test point. The CES method utilizes the best model for each possible value of y, which is identified by the lower envelope of these three pinball loss functions. In this case, the lower envelope has a single finite knot at k_2.

After pre-training and storing T candidate models, namely $M_{\beta,t_1}, \ldots, M_{\beta,t_T}$ for some sub-sequence $(t_1, \ldots, t_T)$ of $[t^{\max}]$, consider the following optimization problem,

$$\hat{M}_{\beta,\text{ces}}(X_{n+1}, y) = \underset{M_{\beta,t_j}\,:\,1 \le j \le T}{\arg\min} \mathcal{L}^{+1}_{\text{es-cal}}(M_{\beta,t_j}, X_{n+1}, y). \tag{A30}$$

This problem is equivalent to identifying the lower envelope of a family of shifted pinball loss functions, similarly to Section 2.5; see Figure A9 for a schematic visualization. Again, this lower envelope can be found at computational cost $\mathcal{O}(T \log T)$, with the same divide-and-conquer algorithm described in Appendix A5. In particular, $\hat{M}_{\beta,\text{ces}}(X_{n+1}, y)$ is a

step function with respect to y with L distinct steps, for some $L = \mathcal{O}(T \log T)$, and it can be written as:

$$\hat{M}_{\beta,\text{ces}}(X_{n+1}, y) = \sum_{l=1}^{L} m_{\beta,l}(X_{n+1}) \mathbb{1}\left[y \in (k_{l-1}^{\beta}, k_l^{\beta}]\right], \tag{A31}$$

where $m_{\beta,l}(X_{n+1}) \in [T]$ represents the best model selected within the interval $(k_{l-1}^{\beta}, k_l^{\beta}]$ such that $m_{\beta,l}(X_{n+1}) \neq m_{\beta,l-1}(X_{n+1})$ for all $l \in [L]$. Above, $k_1^{\beta} \leq k_2^{\beta} \leq \cdots \leq k_L^{\beta}$ denote the *knots* of $\hat{M}_{\beta,\text{ces}}(X_{n+1}, y)$, which also depend on X_{n+1} and are defined as the boundaries in the domain of y between each consecutive pair of steps, with the understanding that $k_0^{\beta} = -\infty$ and $k_{L+1}^{\beta} = +\infty$; see Figure A9 for a schematic visualization.

After computing $\hat{M}_{\beta,\text{ces}}(X_{n+1}, y)$ in (A30) for both β_{low} and β_{high}, we concatenate the respective knots $k_1^{\text{low}}, \ldots, k_{L_1}^{\text{low}}, k_1^{\text{high}}, \ldots, k_{L_2}^{\text{high}}$ and sort them into $k_1 \leq k_2 \leq k_{L_1+L_2}$, so that within each interval $\mathcal{B}_l = (k_{l-1}, k_l]$ for step $l \in [L_1 + L_2]$, there exist exactly one best model for β_{low} and exactly one best model for β_{high}. Then, for each interval $\mathcal{B}_l = (k_{l-1}, k_l]$ associated with step $l \in [L_1 + L_2]$, evaluate the nonconformity score $\hat{E}_i(X_{n+1}, \mathcal{B}_l)$ for all $i \in \mathcal{D}_{\text{es-cal}}$, based on the regression model indicated by $m_{\beta_{\text{low}},l}(X_{n+1})$ and $m_{\beta_{\text{high}},l}(X_{n+1})$; i.e.,

$$\hat{E}_i(X_{n+1}, \mathcal{B}_l) = \max\left\{\hat{q}_{m_{\beta_{\text{low}},l}(X_{n+1})}(X_i) - Y_i, Y_i - \hat{q}_{m_{\beta_{\text{high}},l}(X_{n+1})}(X_i)\right\}. \tag{A32}$$

Let $\hat{Q}_{1-\alpha}(X_{n+1}, \mathcal{B}_l)$ denote the $\lceil (1-\alpha)(1 + |\mathcal{D}_{\text{es-cal}}|)\rceil$-th smallest value among all nonconformity scores $\hat{E}_i(X_{n+1}, \mathcal{B}_l)$, assuming for simplicity that there are no ties; otherwise, ties can be broken at random. Then, define the interval $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)$ as that obtained by applying the conformal prediction method of Romano et al. (2019) with nonconformity scores (A32) based on the estimated conditional quantiles $\hat{q}_{m_{\beta_{\text{low}},l}(X_{n+1})}(X_{n+1})$ and $\hat{q}_{m_{\beta_{\text{high}},l}(X_{n+1})}(X_{n+1})$; that is,

$$\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l) = [\hat{q}_{m_{\beta_{\text{low}},l}(X_{n+1})}(X_{n+1}) - \hat{Q}_{1-\alpha}(X_{n+1}, \mathcal{B}_l), \hat{q}_{m_{\beta_{\text{high}},l}(X_{n+1})}(X_{n+1}) + \hat{Q}_{1-\alpha}(X_{n+1}, \mathcal{B}_l)]. \tag{A33}$$

Finally, the output prediction interval $\hat{C}_\alpha(X_{n+1})$ is given by:

$$\hat{C}_\alpha(X_{n+1}) = \text{Convex}\left(\cup_{l=1}^{L}\{\mathcal{B}_l \cap \hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)\}\right), \tag{A34}$$

where Convex$(\cdot)$ denotes the convex hull of a set. This procedure is summarized in Algorithm A11 and it is guaranteed to produce prediction sets with valid marginal coverage.

Algorithm A11 Conformalized early stopping for quantile regression

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with outcomes $Y_i \in \mathbb{R}$.
Test point with features X_{n+1}. Desired coverage level $1 - \alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
Quantile regression model trainable via (stochastic) gradient descent. Target quantiles $[\beta_{\text{low}}, \beta_{\text{high}}]$.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train for t^{max} epochs and save the intermediate models $M_{\beta_{\text{low}},t_1}, \ldots, M_{\beta_{\text{low}},t_T}, M_{\beta_{\text{high}},t_1}, \ldots, M_{\beta_{\text{high}},t_T}$.
Evaluate $\hat{M}_{\beta_{\text{low}},\text{ces}}(X_{n+1}, y)$ and $\hat{M}_{\beta_{\text{high}},\text{ces}}(X_{n+1}, y)$ as in (A31), using Algorithm A9.
Partition the domain of Y into $L_1 + L_2$ intervals $\mathcal{B}_l$, for $l \in [L_1 + L_2]$, based on the knots of $\hat{M}_{\beta_{\text{low}},\text{ces}}(X_{n+1}, y)$ and $\hat{M}_{\beta_{\text{high}},\text{ces}}(X_{n+1}, y)$.
for $l \in [L_1 + L_2]$ **do**
  Evaluate nonconformity scores $\hat{E}_i(X_{n+1}, \mathcal{B}_l)$ for all $i \in \mathcal{D}_{\text{es-cal}}$ as in (A32).
  Compute $\hat{Q}_{1-\alpha}(X_{n+1}, \mathcal{B}_l)$ as the $\lceil (1-\alpha)(1 + |\mathcal{D}_{\text{es-cal}}|)\rceil$-th largest value among $\hat{E}_i(X_{n+1}, \mathcal{B}_l)$.
  Construct the interval $\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)$ according to (A33).
end for
Output: Prediction interval $\hat{C}_\alpha(X_{n+1})$ given as a function of $\{\hat{C}_\alpha(X_{n+1}, \mathcal{B}_l)\}_{l=1}^{L}$ by (A34).

Theorem A2. *Assume $(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})$ are exchangeable, and let $\hat{C}_\alpha(X_{n+1})$ be the output of Algorithm A11, as given by (A34), for any given $\alpha \in (0, 1)$. Then, $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})] \geq 1 - \alpha$.*

A7.3. Avoiding Empty Predictions

Similarly to Section 2.5, it is possible (although unlikely) that Algorithm A11 may sometimes produce an empty prediction set. Therefore, we presents Algorithm A12, which extends Algorithm A11 in such a way as to explicitly avoid returning empty prediction intervals. As the intervals given by Algorithm A12 always contain those output by Algorithm A11, it follows from Theorem A2 that Algorithm A12 also enjoys guaranteed coverage; see Corollary A3.

Algorithm A12 Conformalized early stopping for quantile regression, avoiding empty predictions

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with outcomes $Y_i \in \mathbb{R}$.
Test point with features X_{n+1}. Desired coverage level $1 - \alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
Quantile regression model trainable via (stochastic) gradient descent. Target quantiles $[\beta_{\text{low}}, \beta_{\text{high}}]$.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train for t^{max} epochs and save the intermediate models $M_{\beta_{\text{low}},t_1}, \ldots, M_{\beta_{\text{low}},t_T}, M_{\beta_{\text{high}},t_1}, \ldots, M_{\beta_{\text{high}},t_T}$.
Evaluate $\hat{C}_\alpha(X_{n+1})$ using Algorithm A11.
if $\hat{C}_\alpha(X_{n+1}) = \emptyset$ **then**
  Evaluate $\hat{C}^{\text{naive}}_\alpha(X_{n+1})$ using Algorithm A13. Set $\hat{C}_\alpha(X_{n+1}) = \hat{C}^{\text{naive}}_\alpha(X_{n+1})$.
end if
Output: A non-empty prediction interval $\hat{C}_\alpha(X_{n+1})$.

Corollary A3. *Assume $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, Y_{n+1})$ are exchangeable random samples, and let $\hat{C}_\alpha(X_{n+1})$ be the output of Algorithm A12, for any $\alpha \in (0, 1)$. Then, $\mathbb{P}[Y_{n+1} \in \hat{C}_\alpha(X_{n+1})] \geq 1 - \alpha$.*

A7.4. Implementation of the Naive Benchmark

Algorithm A13 Naive conformal quantile regression benchmark with greedy early stopping

Input: Exchangeable data points $(X_1, Y_1), \ldots, (X_n, Y_n)$ with outcomes $Y_i \in \mathbb{R}$.
Test point with features X_{n+1}. Desired coverage level $1 - \alpha$.
Maximum number of training epochs t^{max}; storage period hyper-parameter τ.
Quantile regression model trainable via (stochastic) gradient descent minimizing the pinball loss.
Target quantiles $[\beta_{\text{low}}, \beta_{\text{high}}]$.
Randomly split the exchangeable data points into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{es-cal}}$.
Train for t^{max} epochs and save the intermediate models $M_{\beta_{\text{low}},t_1}, \ldots, M_{\beta_{\text{low}}t_T}, M_{\beta_{\text{high}},t_1}, \ldots, M_{\beta_{\text{high}}t_T}$.
Pick the most promising models $t^*_{\text{low}}, t^*_{\text{high}} \in [T]$ minimizing $\mathcal{L}_{\text{es-cal}}(M_t)$ in (A29).
Evaluate nonconformity scores $\hat{E}_i(X_{n+1}) = \max\{\hat{q}_{t^*_{\text{low}}}(X_i) - Y_i, Y_i - \hat{q}_{t^*_{\text{high}}}(X_i)\}$ for all $i \in \mathcal{D}_{\text{es-cal}}$.
Compute $\hat{Q}_{1-\alpha}(X_{n+1}) = \lceil (1-\alpha)(1 + |\mathcal{D}_{\text{es-cal}}|) \rceil$-th largest value in $\hat{E}_i(X_{n+1})$ for $i \in \mathcal{D}_{\text{es-cal}}$.
Output: Naive prediction interval $\hat{C}^{\text{naive}}_\alpha(X_{n+1}) = [\hat{q}_{t^*_{\text{low}}}(X_{n+1}) - \hat{Q}_{1-\alpha}(X_{n+1}), \hat{q}_{t^*_{\text{high}}}(X_{n+1}) + \hat{Q}_{1-\alpha}(X_{n+1})]$.

A8. Additional Results from Numerical Experiments

A8.1. Outlier Detection, Classification, and Regression

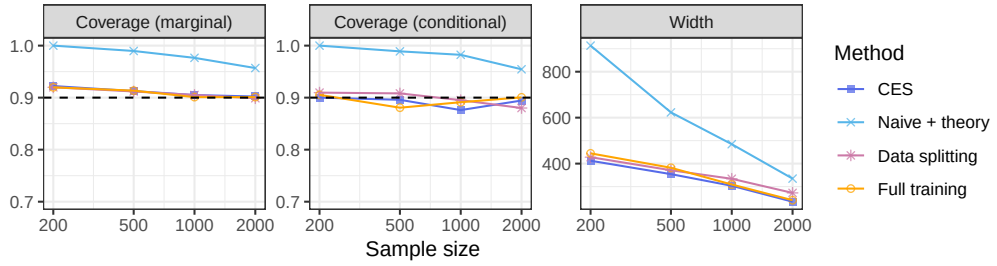


Figure A10. Performance of conformal prediction intervals based on regression models trained with different methods, on the *bike* data set (dat, a). The results are shown as a function of the total sample size. The nominal marginal coverage level is 90%. See Table A4 for additional details and standard errors.

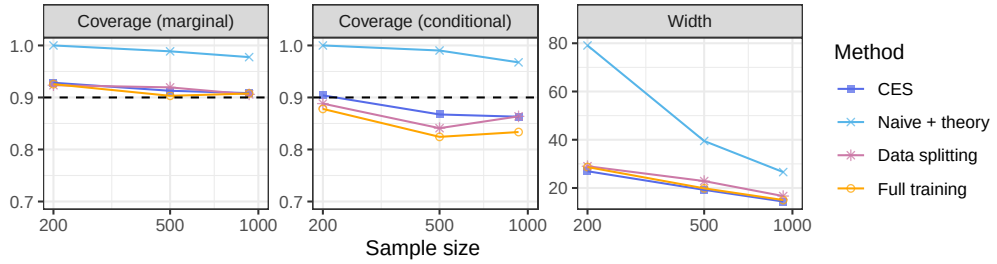


Figure A11. Performance of conformal prediction intervals based on regression models trained with different methods, on the *concrete* data set (dat, c). The results are shown as a function of the total sample size. The nominal marginal coverage level is 90%. See Table A5 for additional details and standard errors.

Table A1. Performance of outlier detection based on classification models trained with different methods, on the *CIFAR10* data set (Krizhevsky, 2009). Other details are as in Figure 4. The numbers in parenthesis indicate standard errors. The numbers in bold highlight TPR values within 1 standard error of the best TPR across all methods, for each sample size.

Sample size	Method	TPR	FPR
500			
500	CES	**0.296 (0.008)**	0.098 (0.003)
500	Naive	**0.295 (0.008)**	0.097 (0.003)
500	Naive + theory	0.114 (0.006)	0.016 (0.001)
500	Data splitting	0.234 (0.008)	0.091 (0.003)
500	Full training	0.217 (0.011)	0.072 (0.004)
1000			
1000	CES	**0.401 (0.007)**	0.100 (0.004)
1000	Naive	**0.401 (0.007)**	0.100 (0.004)
1000	Naive + theory	0.237 (0.006)	0.030 (0.002)
1000	Data splitting	0.337 (0.009)	0.094 (0.003)
1000	Full training	0.189 (0.013)	0.055 (0.004)
2000			
2000	CES	**0.450 (0.005)**	0.100 (0.003)
2000	Naive	**0.450 (0.005)**	0.100 (0.003)
2000	Naive + theory	0.337 (0.006)	0.048 (0.002)
2000	Data splitting	0.404 (0.007)	0.095 (0.003)
2000	Full training	0.050 (0.009)	0.017 (0.003)

Table A2. Performance of multi-class classification based on classification models trained with different methods, on the *CIFAR10* data set (Krizhevsky, 2009). Other details are as in Figure 5. The numbers in parenthesis indicate standard errors. The numbers in bold highlight cardinality values within 1 standard error of the best cardinality across all methods, for each sample size.

Sample size	Method	Cardinality	Marignal coverage
500			
500	CES	**6.754 (0.074)**	0.908 (0.003)
500	Naive	**6.735 (0.072)**	0.906 (0.003)
500	Naive + theory	9.193 (0.052)	0.988 (0.001)
500	Data splitting	7.022 (0.077)	0.900 (0.004)
500	Full training	**6.759 (0.091)**	0.909 (0.004)
1000			
1000	CES	**5.902 (0.060)**	0.902 (0.003)
1000	Naive	**5.908 (0.059)**	0.901 (0.003)
1000	Naive + theory	7.767 (0.064)	0.972 (0.002)
1000	Data splitting	6.294 (0.063)	0.900 (0.004)
1000	Full training	6.270 (0.092)	0.897 (0.004)
2000			
2000	CES	**5.352 (0.045)**	0.903 (0.003)
2000	Naive	**5.347 (0.045)**	0.902 (0.003)
2000	Naive + theory	6.609 (0.049)	0.955 (0.002)
2000	Data splitting	5.674 (0.040)	0.904 (0.003)
2000	Full training	7.776 (0.194)	0.934 (0.006)

Table A3. Performance of conformal prediction intervals based on regression models trained with different methods, on the *bio* data set (dat, b). Other details are as in Figure 2. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	bio	CES	**18.740 (0.123)**	0.924 (0.005)	0.898 (0.014)
200	bio	Naive	**18.544 (0.133)**	0.917 (0.005)	0.899 (0.015)
200	bio	Naive + theory	20.942 (0.006)	1.000 (0.000)	1.000 (0.000)
200	bio	Data splitting	19.068 (0.113)	0.925 (0.005)	0.902 (0.015)
200	bio	Full training	**18.673 (0.125)**	0.919 (0.004)	0.890 (0.018)
500					
500	bio	CES	**17.435 (0.125)**	0.914 (0.004)	0.909 (0.013)
500	bio	Naive	**17.363 (0.134)**	0.910 (0.004)	0.890 (0.017)
500	bio	Naive + theory	20.391 (0.061)	0.993 (0.001)	0.996 (0.001)
500	bio	Data splitting	18.076 (0.123)	0.911 (0.004)	0.888 (0.015)
500	bio	Full training	18.245 (0.129)	0.918 (0.003)	0.890 (0.019)
1000					
1000	bio	CES	**16.251 (0.081)**	0.901 (0.003)	0.885 (0.015)
1000	bio	Naive	**16.219 (0.084)**	0.900 (0.003)	0.890 (0.012)
1000	bio	Naive + theory	19.167 (0.073)	0.977 (0.002)	0.976 (0.006)
1000	bio	Data splitting	16.728 (0.089)	0.903 (0.003)	0.887 (0.016)
1000	bio	Full training	17.962 (0.141)	0.902 (0.004)	0.814 (0.022)
2000					
2000	bio	CES	**15.812 (0.042)**	0.899 (0.003)	0.893 (0.015)
2000	bio	Naive	**15.805 (0.042)**	0.899 (0.003)	0.897 (0.014)
2000	bio	Naive + theory	17.691 (0.047)	0.957 (0.002)	0.963 (0.006)
2000	bio	Data splitting	16.043 (0.059)	0.900 (0.003)	0.903 (0.015)
2000	bio	Full training	17.014 (0.113)	0.902 (0.003)	0.829 (0.019)

Table A4. Performance of conformal prediction intervals based on regression models trained with different methods, on the *bike* data set (dat, a). Other details are as in Figure A10. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	bike	CES	412.534 (6.439)	0.922 (0.004)	0.900 (0.018)
200	bike	Naive	**392.474 (6.019)**	0.908 (0.005)	0.878 (0.016)
200	bike	Naive + theory	913.440 (3.737)	1.000 (0.000)	1.000 (0.000)
200	bike	Data splitting	427.964 (7.282)	0.920 (0.004)	0.910 (0.016)
200	bike	Full training	444.656 (6.760)	0.920 (0.005)	0.905 (0.013)
500					
500	bike	CES	354.180 (4.183)	0.913 (0.004)	0.896 (0.017)
500	bike	Naive	**343.641 (4.220)**	0.902 (0.004)	0.889 (0.018)
500	bike	Naive + theory	622.837 (8.381)	0.990 (0.001)	0.989 (0.004)
500	bike	Data splitting	371.079 (3.777)	0.912 (0.004)	0.908 (0.014)
500	bike	Full training	381.951 (5.175)	0.913 (0.004)	0.881 (0.018)
1000					
1000	bike	CES	**303.516 (3.047)**	0.905 (0.004)	0.876 (0.017)
1000	bike	Naive	**300.091 (3.127)**	0.903 (0.004)	0.894 (0.015)
1000	bike	Naive + theory	484.565 (4.958)	0.977 (0.002)	0.982 (0.003)
1000	bike	Data splitting	333.939 (2.891)	0.905 (0.003)	0.895 (0.018)
1000	bike	Full training	308.981 (3.760)	0.901 (0.004)	0.891 (0.016)
2000					
2000	bike	CES	**234.322 (1.935)**	0.902 (0.003)	0.894 (0.018)
2000	bike	Naive	**231.571 (1.956)**	0.897 (0.003)	0.893 (0.018)
2000	bike	Naive + theory	334.724 (2.988)	0.957 (0.002)	0.954 (0.012)
2000	bike	Data splitting	272.589 (2.532)	0.899 (0.003)	0.880 (0.019)
2000	bike	Full training	240.714 (2.389)	0.901 (0.003)	0.901 (0.015)

Table A5. Performance of conformal prediction intervals based on regression models trained with different methods, on the *concrete* data set (dat, c). Other details are as in Figure A11. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	concrete	CES	26.948 (0.515)	0.928 (0.005)	0.904 (0.016)
200	concrete	Naive	**24.793 (0.520)**	0.903 (0.006)	0.856 (0.021)
200	concrete	Naive + theory	79.089 (0.130)	1.000 (0.000)	1.000 (0.000)
200	concrete	Data splitting	29.021 (0.564)	0.924 (0.004)	0.888 (0.018)
200	concrete	Full training	28.676 (0.568)	0.926 (0.005)	0.878 (0.021)
500					
500	concrete	CES	19.232 (0.263)	0.913 (0.004)	0.867 (0.018)
500	concrete	Naive	**18.340 (0.276)**	0.896 (0.004)	0.829 (0.022)
500	concrete	Naive + theory	39.492 (0.861)	0.989 (0.002)	0.990 (0.003)
500	concrete	Data splitting	22.876 (0.323)	0.919 (0.003)	0.841 (0.023)
500	concrete	Full training	19.857 (0.300)	0.903 (0.004)	0.824 (0.024)
930					
930	concrete	CES	14.399 (0.134)	0.908 (0.002)	0.863 (0.014)
930	concrete	Naive	**13.738 (0.127)**	0.899 (0.002)	0.863 (0.011)
930	concrete	Naive + theory	26.596 (0.334)	0.978 (0.001)	0.967 (0.004)
930	concrete	Data splitting	16.659 (0.122)	0.906 (0.002)	0.864 (0.014)
930	concrete	Full training	14.998 (0.143)	0.908 (0.002)	0.834 (0.016)

A8.2. Quantile Regression

We apply the CES quantile regression method extensively to the following publicly available and commonly investigated regression data sets from the UCI Machine Learning repository (Pınar Tüfekci, 2012): medical expenditure panel survey number 21 (*MEPS_21*) (mep); blog feedback (*blog_data*) (blo); Tennessee's student teacher achievement ratio (*STAR*) (Achilles et al., 2008); community and crimes (*community*) (com); physicochemical properties of protein tertiary structure (*bio*) (dat, b); house sales in King County (*homes*) (hom); and hourly and daily counts of rental bikes (*bike*) (dat, a). These data sets were previously also considered by Romano et al. (2019).

As in the previous sections, we compare CES to the usual three benchmarks, now implemented based on quantile regression: naive early stopping with the *hybrid* theoretical correction for the nominal coverage level, early stopping based on data splitting and full model training without early stopping. We follow the same model architecture and data preprocessing steps as in Romano et al. (2019). To be specific, the input features are standardized to have zero mean and unit variance, and the response values are rescaled by diving the absolute mean of the training responses. All methods utilize the same neural network with three hidden layers and ReLU activation functions between layers, trained for up to 2000 epochs. The parameters are trained minimizing the pinball loss function A24 with Adam optimizer (Kingma & Ba, 2014), minibatches of size 25, 0 weight decay and dropout, and fixed learning rate (0.001 for *STAR, homes, bike, and bio*, 0.0001 for *community*, and 0.00005 for *MEPS_21 and blog_data*).

The models are calibrated in such a way as to produce conformal prediction sets with guaranteed 90% marginal coverage for a test set of 1000 independent data points. The total sample size available for training, early stopping and calibration is varied between 200 and 2000 (200 and 1000 for small data sets such as *community* and *STAR*). These data are allocated for specific training, early-stopping, and calibration operations as in Sections 3.1–3.2. Again, the performance of each method is measured in terms of marginal coverage, worst-slab conditional coverage (Cauchois et al., 2021), and average width of the prediction intervals. All results are averaged over 25 independent experiments, each based on a different random sample from the original raw data sets.

Figure A12 summarizes the performance of the four alternative methods on the *homes* data, as a function of the total sample size; The error bar corresponding to standard errors are plotted around each data point. These results show that all methods reach 90% marginal coverage in practice, as anticipated by the mathematical guarantees, although the theoretical correction for the naive early stopping method appears to be overly conservative. Full training, though producing the smallest prediction bands, has very low conditional coverage, which indicates that fully trained neural network models can suffer from overfitting and therefore is not appealing. Data splitting method beats full training as it gives higher approximated conditional coverage, and CES further beats data splitting in terms of conditional coverage, meanwhile producing prediction intervals of similar length as data splitting. These patterns hold true in general for additional data sets, as illustrated by Figures A13–A18 and by Tables A6–A12. Tables A6–A12 also include the results obtained with the naive benchmark applied without the necessary theoretical correction, which performs similarly to CES.

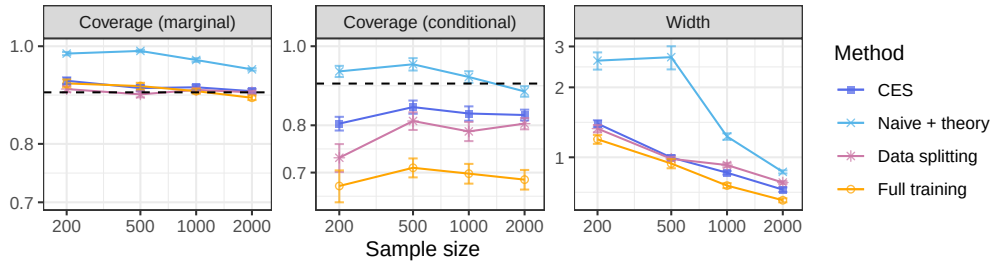


Figure A12. Average performance, as a function of the sample size, of conformal prediction sets for quantile regression based on neural networks trained and calibrated with different methods, on the *homes* data (hom). The marginal coverage is theoretically guaranteed to be above 90%. Ideally, the conditional coverage should high and the prediction intervals should be tight. See Table A6 for additional details and standard errors.

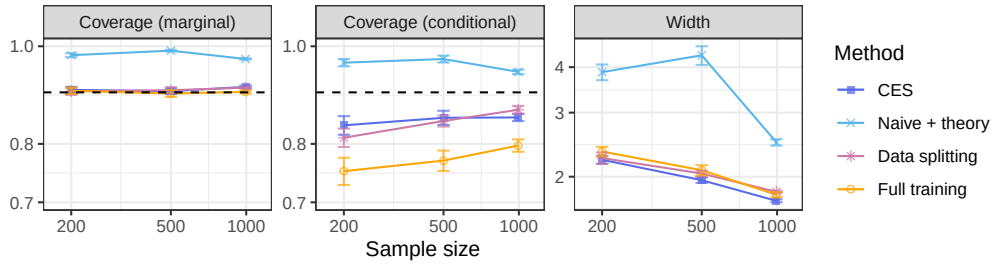


Figure A13. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *community* data set (com). The results are shown as a function of the total sample size with error bars corresponding to standard error. The nominal marginal coverage level is 90%. See Table A7 for additional details and standard errors.

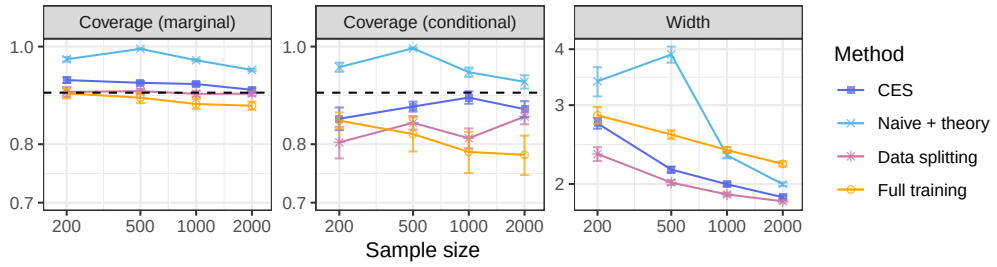


Figure A14. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *bio* data set (dat, b). The results are shown as a function of the total sample size with error bars corresponding to standard error. The nominal marginal coverage level is 90%. See Table A8 for additional details and standard errors.

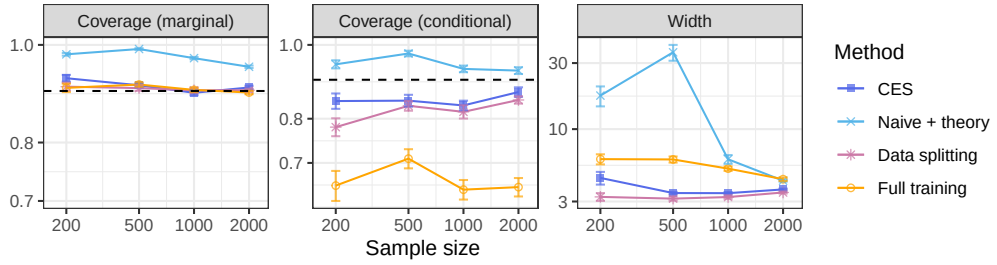


Figure A15. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *meps_21* data set (mep).The results are shown as a function of the total sample size with error bars corresponding to standard error. The nominal marginal coverage level is 90%. See Table A9 for additional details and standard errors.

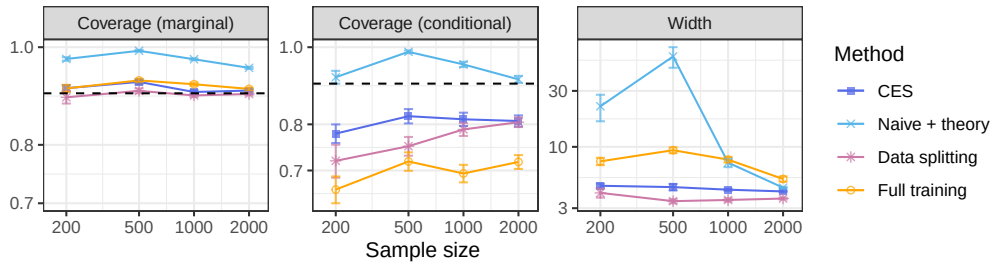


Figure A16. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *blog_data* data set (blo). The results are shown as a function of the total sample size with error bars corresponding to standard error. The nominal marginal coverage level is 90%. See Table A10 for additional details and standard errors.

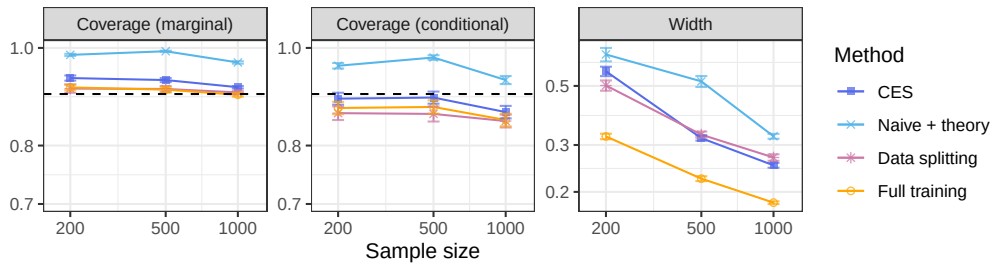


Figure A17. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *STAR* data set (Achilles et al., 2008). The results are shown as a function of the total sample size with error bars corresponding to standard error. The nominal marginal coverage level is 90%. See Table A11 for additional details and standard errors.

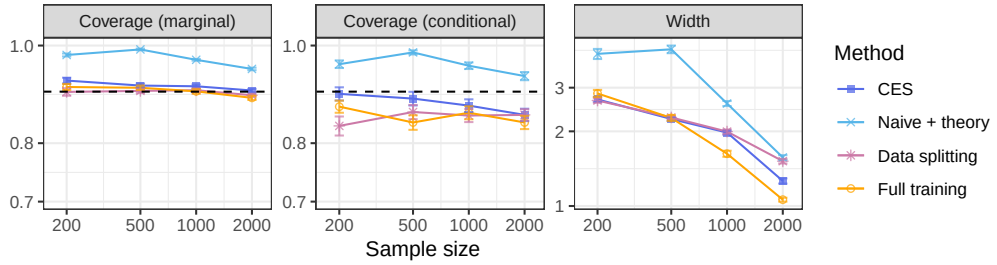


Figure A18. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *bike* data set (dat, a). The results are shown as a function of the total sample size with error bars corresponding to standard error. The nominal marginal coverage level is 90%. See Table A12 for additional details and standard errors.

Table A6. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *homes* data set (hom). Other details are as in Figure A12. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	homes	CES	1.396 (0.047)	0.924 (0.007)	0.804 (0.016)
200	homes	Naive	**1.171 (0.048)**	0.904 (0.009)	0.761 (0.029)
200	homes	Naive + theory	2.607 (0.222)	0.983 (0.004)	0.931 (0.015)
200	homes	Data splitting	1.329 (0.049)	0.907 (0.006)	0.730 (0.029)
200	homes	Full training	**1.195 (0.053)**	0.919 (0.008)	0.674 (0.030)
500					
500	homes	CES	0.997 (0.022)	0.909 (0.007)	0.842 (0.016)
500	homes	Naive	**0.940 (0.026)**	0.899 (0.007)	0.798 (0.018)
500	homes	Naive + theory	2.698 (0.308)	0.989 (0.003)	0.950 (0.017)
500	homes	Data splitting	0.985 (0.019)	0.896 (0.006)	0.810 (0.020)
500	homes	Full training	**0.940 (0.043)**	0.913 (0.007)	0.710 (0.019)
1000					
1000	homes	CES	0.858 (0.015)	0.910 (0.005)	0.827 (0.017)
1000	homes	Naive	0.824 (0.017)	0.894 (0.006)	0.784 (0.015)
1000	homes	Naive + theory	1.229 (0.039)	0.969 (0.004)	0.917 (0.015)
1000	homes	Data splitting	0.926 (0.020)	0.905 (0.004)	0.786 (0.021)
1000	homes	Full training	**0.755 (0.019)**	0.902 (0.005)	0.698 (0.019)
2000					
2000	homes	CES	0.726 (0.016)	0.902 (0.005)	0.824 (0.013)
2000	homes	Naive	0.667 (0.013)	0.891 (0.003)	0.799 (0.016)
2000	homes	Naive + theory	0.865 (0.013)	0.949 (0.003)	0.880 (0.013)
2000	homes	Data splitting	0.779 (0.011)	0.900 (0.003)	0.804 (0.013)
2000	homes	Full training	**0.653 (0.013)**	0.889 (0.005)	0.686 (0.019)

Table A7. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *community* data set (com). Other details are as in Figure A13. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	community	CES	**2.226 (0.057)**	0.905 (0.006)	0.835 (0.018)
200	community	Naive	**2.100 (0.065)**	0.891 (0.009)	0.814 (0.018)
200	community	Naive + theory	3.876 (0.193)	0.980 (0.004)	0.963 (0.008)
200	community	Data splitting	**2.250 (0.081)**	0.902 (0.007)	0.811 (0.017)
200	community	Full training	2.345 (0.070)	0.903 (0.007)	0.752 (0.023)
500					
500	community	CES	**1.957 (0.033)**	0.902 (0.006)	0.849 (0.014)
500	community	Naive	**1.866 (0.043)**	0.889 (0.007)	0.830 (0.012)
500	community	Naive + theory	4.312 (0.254)	0.990 (0.002)	0.971 (0.008)
500	community	Data splitting	2.041 (0.037)	0.904 (0.006)	0.843 (0.011)
500	community	Full training	2.084 (0.067)	0.898 (0.007)	0.770 (0.018)
994					
994	community	CES	**1.717 (0.016)**	0.911 (0.003)	0.850 (0.007)
994	community	Naive	**1.601 (0.019)**	0.902 (0.003)	0.853 (0.008)
994	community	Naive + theory	2.486 (0.049)	0.971 (0.002)	0.943 (0.005)
994	community	Data splitting	1.818 (0.013)	0.910 (0.002)	0.865 (0.008)
994	community	Full training	1.784 (0.029)	0.901 (0.003)	0.797 (0.011)

Table A8. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *bio* data set (dat, b). Other details are as in Figure A14. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	bio	CES	2.732 (0.074)	0.926 (0.006)	0.848 (0.022)
200	bio	Naive	**2.018 (0.064)**	0.876 (0.011)	0.805 (0.025)
200	bio	Naive + theory	3.392 (0.254)	0.971 (0.005)	0.954 (0.010)
200	bio	Data splitting	**2.334 (0.083)**	0.902 (0.010)	0.803 (0.029)
200	bio	Full training	2.848 (0.124)	0.899 (0.010)	0.844 (0.016)
500					
500	bio	CES	2.154 (0.034)	0.921 (0.004)	0.872 (0.010)
500	bio	Naive	**1.900 (0.032)**	0.892 (0.007)	0.839 (0.014)
500	bio	Naive + theory	3.893 (0.161)	0.995 (0.001)	0.996 (0.002)
500	bio	Data splitting	**2.015 (0.024)**	0.903 (0.004)	0.840 (0.013)
500	bio	Full training	2.579 (0.057)	0.890 (0.011)	0.819 (0.032)
1000					
1000	bio	CES	1.997 (0.019)	0.918 (0.004)	0.890 (0.013)
1000	bio	Naive	**1.871 (0.018)**	0.894 (0.005)	0.838 (0.018)
1000	bio	Naive + theory	2.318 (0.033)	0.969 (0.002)	0.943 (0.010)
1000	bio	Data splitting	**1.895 (0.018)**	0.898 (0.006)	0.811 (0.018)
1000	bio	Full training	2.381 (0.035)	0.877 (0.010)	0.786 (0.037)
2000					
2000	bio	CES	1.869 (0.018)	0.906 (0.004)	0.867 (0.016)
2000	bio	Naive	**1.763 (0.017)**	0.890 (0.004)	0.834 (0.016)
2000	bio	Naive + theory	1.998 (0.017)	0.948 (0.003)	0.922 (0.014)
2000	bio	Data splitting	**1.830 (0.013)**	0.898 (0.004)	0.852 (0.015)
2000	bio	Full training	2.219 (0.029)	0.874 (0.008)	0.781 (0.035)

Table A9. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *MEPS_21* data set (mep). Other details are as in Figure A15. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	meps_21	CES	4.442 (0.468)	0.927 (0.007)	0.844 (0.020)
200	meps_21	Naive	**3.602 (0.452)**	0.901 (0.010)	0.786 (0.025)
200	meps_21	Naive + theory	17.470 (2.874)	0.979 (0.003)	0.943 (0.012)
200	meps_21	Data splitting	**3.239 (0.212)**	0.909 (0.008)	0.780 (0.021)
200	meps_21	Full training	6.070 (0.509)	0.906 (0.009)	0.654 (0.030)
500					
500	meps_21	CES	3.453 (0.074)	0.912 (0.004)	0.845 (0.015)
500	meps_21	Naive	**3.077 (0.068)**	0.898 (0.005)	0.802 (0.012)
500	meps_21	Naive + theory	35.882 (4.705)	0.991 (0.002)	0.974 (0.008)
500	meps_21	Data splitting	**3.147 (0.109)**	0.906 (0.006)	0.832 (0.012)
500	meps_21	Full training	6.038 (0.305)	0.913 (0.005)	0.709 (0.021)
1000					
1000	meps_21	CES	3.447 (0.068)	0.896 (0.005)	0.833 (0.012)
1000	meps_21	Naive	**3.102 (0.073)**	0.888 (0.006)	0.825 (0.014)
1000	meps_21	Naive + theory	6.054 (0.443)	0.970 (0.002)	0.930 (0.009)
1000	meps_21	Data splitting	**3.231 (0.093)**	0.901 (0.005)	0.817 (0.016)
1000	meps_21	Full training	5.183 (0.188)	0.902 (0.004)	0.646 (0.019)
2000					
2000	meps_21	CES	3.666 (0.054)	0.907 (0.004)	0.867 (0.012)
2000	meps_21	Naive	**3.454 (0.047)**	0.902 (0.004)	0.875 (0.013)
2000	meps_21	Naive + theory	4.242 (0.069)	0.951 (0.003)	0.925 (0.010)
2000	meps_21	Data splitting	**3.485 (0.060)**	0.902 (0.003)	0.847 (0.009)
2000	meps_21	Full training	4.349 (0.128)	0.897 (0.003)	0.651 (0.018)

Table A10. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *blog_data* data set (blo). Other details are as in Figure A16. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	blog_data	CES	4.658 (0.194)	0.911 (0.007)	0.779 (0.021)
200	blog_data	Naive	**3.427 (0.210)**	0.897 (0.008)	0.766 (0.021)
200	blog_data	Naive + theory	22.166 (5.645)	0.973 (0.004)	0.917 (0.017)
200	blog_data	Data splitting	**4.049 (0.359)**	0.892 (0.013)	0.720 (0.034)
200	blog_data	Full training	7.524 (0.544)	0.910 (0.007)	0.662 (0.026)
500					
500	blog_data	CES	4.538 (0.291)	0.924 (0.005)	0.819 (0.017)
500	blog_data	Naive	3.986 (0.333)	0.917 (0.005)	0.796 (0.014)
500	blog_data	Naive + theory	58.965 (11.745)	0.992 (0.002)	0.987 (0.004)
500	blog_data	Data splitting	**3.443 (0.139)**	0.905 (0.005)	0.751 (0.020)
500	blog_data	Full training	9.364 (0.535)	0.927 (0.003)	0.719 (0.019)
1000					
1000	blog_data	CES	4.313 (0.088)	0.903 (0.004)	0.812 (0.016)
1000	blog_data	Naive	**3.559 (0.073)**	0.896 (0.004)	0.816 (0.016)
1000	blog_data	Naive + theory	7.336 (0.620)	0.973 (0.002)	0.951 (0.007)
1000	blog_data	Data splitting	**3.525 (0.119)**	0.896 (0.003)	0.788 (0.015)
1000	blog_data	Full training	7.784 (0.427)	0.919 (0.003)	0.694 (0.018)
2000					
2000	blog_data	CES	4.169 (0.066)	0.906 (0.003)	0.808 (0.013)
2000	blog_data	Naive	**3.671 (0.047)**	0.899 (0.003)	0.783 (0.018)
2000	blog_data	Naive + theory	4.486 (0.074)	0.954 (0.002)	0.911 (0.009)
2000	blog_data	Data splitting	**3.625 (0.069)**	0.899 (0.003)	0.805 (0.011)
2000	blog_data	Full training	5.323 (0.274)	0.909 (0.003)	0.718 (0.014)

Table A11. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *STAR* data set (Achilles et al., 2008). Other details are as in Figure A17. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	star	CES	0.567 (0.023)	0.933 (0.006)	0.891 (0.012)
200	star	Naive	0.421 (0.031)	0.904 (0.008)	0.847 (0.012)
200	star	Naive + theory	0.656 (0.038)	0.984 (0.002)	0.960 (0.006)
200	star	Data splitting	0.502 (0.022)	0.911 (0.009)	0.862 (0.013)
200	star	Full training	**0.323 (0.008)**	0.914 (0.006)	0.872 (0.012)
500					
500	star	CES	0.318 (0.007)	0.929 (0.004)	0.893 (0.012)
500	star	Naive	0.288 (0.008)	0.902 (0.007)	0.848 (0.016)
500	star	Naive + theory	0.520 (0.025)	0.992 (0.001)	0.978 (0.005)
500	star	Data splitting	0.328 (0.009)	0.910 (0.006)	0.860 (0.015)
500	star	Full training	**0.224 (0.005)**	0.909 (0.006)	0.874 (0.013)
1000					
1000	star	CES	0.252 (0.005)	0.914 (0.003)	0.864 (0.012)
1000	star	Naive	0.236 (0.007)	0.897 (0.003)	0.862 (0.011)
1000	star	Naive + theory	0.323 (0.007)	0.967 (0.003)	0.929 (0.009)
1000	star	Data splitting	0.269 (0.007)	0.903 (0.003)	0.846 (0.012)
1000	star	Full training	**0.182 (0.002)**	0.898 (0.004)	0.848 (0.012)
1161					
1161	star	CES	0.237 (0.005)	0.920 (0.004)	0.882 (0.012)
1161	star	Naive	0.220 (0.005)	0.900 (0.005)	0.860 (0.012)
1161	star	Naive + theory	0.308 (0.007)	0.969 (0.002)	0.944 (0.007)
1161	star	Data splitting	0.254 (0.007)	0.903 (0.004)	0.864 (0.014)
1161	star	Full training	**0.174 (0.003)**	0.896 (0.004)	0.863 (0.013)

Table A12. Performance of conformal prediction intervals based on quantile regression models trained with different methods, on the *bike* data set (dat, a). Other details are as in Figure A18. The numbers in parenthesis indicate standard errors. The numbers in bold highlight width values within 1 standard error of the best width across all methods, for each sample size. The numbers in red highlight coverage values below 0.85.

				Coverage	
Sample size	Data	Method	Width	Marginal	Conditional
200					
200	bike	CES	**2.690 (0.065)**	0.923 (0.006)	0.895 (0.014)
200	bike	Naive	**2.535 (0.053)**	0.905 (0.008)	0.868 (0.015)
200	bike	Naive + theory	4.106 (0.192)	0.979 (0.003)	0.958 (0.008)
200	bike	Data splitting	**2.663 (0.056)**	0.899 (0.008)	0.832 (0.018)
200	bike	Full training	2.843 (0.095)	0.910 (0.006)	0.870 (0.012)
500					
500	bike	CES	**2.244 (0.039)**	0.913 (0.004)	0.886 (0.014)
500	bike	Naive	**2.137 (0.047)**	0.901 (0.006)	0.880 (0.012)
500	bike	Naive + theory	4.284 (0.152)	0.991 (0.002)	0.984 (0.005)
500	bike	Data splitting	**2.275 (0.032)**	0.902 (0.004)	0.859 (0.014)
500	bike	Full training	**2.265 (0.062)**	0.908 (0.004)	0.839 (0.014)
1000					
1000	bike	CES	1.978 (0.036)	0.911 (0.004)	0.872 (0.013)
1000	bike	Naive	1.779 (0.057)	0.893 (0.004)	0.821 (0.016)
1000	bike	Naive + theory	2.590 (0.061)	0.968 (0.002)	0.955 (0.007)
1000	bike	Data splitting	1.996 (0.040)	0.903 (0.004)	0.852 (0.013)
1000	bike	Full training	**1.625 (0.045)**	0.901 (0.005)	0.858 (0.013)
2000					
2000	bike	CES	1.262 (0.030)	0.902 (0.003)	0.854 (0.012)
2000	bike	Naive	1.113 (0.029)	0.884 (0.004)	0.847 (0.012)
2000	bike	Naive + theory	1.570 (0.033)	0.948 (0.003)	0.933 (0.009)
2000	bike	Data splitting	1.517 (0.033)	0.893 (0.003)	0.853 (0.011)
2000	bike	Full training	**1.061 (0.019)**	0.887 (0.004)	0.839 (0.013)

A9. Mathematical Proofs

Proof of Theorem 2.1. It suffices to show that the nonconformity scores $\hat{S}_i$ for $i \in \{n+1\} \cup \mathcal{D}_{\text{es-cal}}$ are exchangeable. In fact, if the nonconformity scores are almost-surely unique, this implies the rank of $\hat{S}_{n+1}$ is uniformly distributed over $\{\hat{S}_i\}_{i \in \{n+1\} \cup \mathcal{D}_{\text{es-cal}}}$, and in that case the conformal p-value is uniformly distributed over $\{1/(1+|\mathcal{D}_{\text{es-cal}}|), 2/(1+|\mathcal{D}_{\text{es-cal}}|), \ldots, 1\}$. If the nonconformity scores are not almost-surely unique and ties are not broken at random, then the distribution of the conformal p-value becomes stochastically larger than uniform, in which case the result still holds. To prove the exchangeability of the nonconformity scores, let σ be any permutation of $\{n+1\} \cup \mathcal{D}_{\text{es-cal}}$, and imagine applying Algorithm A4, with the same random seed, to the shuffled data set indexed by $\sigma(\{n+1\} \cup \mathcal{D}_{\text{es-cal}})$, which has the same distribution as the original data set. To clarify the notation, we will refer to quantities computed under this data shuffling scenario with their usual symbol followed by an apostrophe; i.e., M'_{t_1} instead of M_{t_1}. As the gradient updates only involve the unperturbed observations in $\mathcal{D}_{\text{train}}$ and the maximum number of epochs $t^{\max}$ is fixed, the sequence of saved models remains exactly the same under this scenario: $(M'_{t_1}, \ldots, M'_{t_T}) = (M_{t_1}, \ldots, M_{t_T})$. Further, the loss function in (1) is also invariant to permutations of $\{n+1\} \cup \mathcal{D}_{\text{es-cal}}$, in the sense that $\mathcal{L}^{+1'}_{\text{es-cal}} = \mathcal{L}^{+1}_{\text{es-cal}}$, because $\mathcal{L}$ is additive. Therefore, the model selected according to (2) is also invariant, $\hat{M}'_{\text{ces}} = \hat{M}_{\text{ces}}$, which implies the nonconformity scores are simply re-ordered: $\hat{S}'_{\sigma(i)} = \hat{S}_i$. Therefore, we have:

$$\begin{aligned}\sigma(\{\hat{S}_i\}_{i \in \{n+1\} \cup \mathcal{D}_{\text{cal}}}) &= \{\hat{S}'_i\}_{i \in \{n+1\} \cup \mathcal{D}_{\text{cal}}} \\ &\overset{d}{=} \{\hat{S}_i\}_{i \in \{n+1\} \cup \mathcal{D}_{\text{cal}}},\end{aligned}$$

where the last equality follows from the initial data exchangeability assumption. □

Proof of Theorem 2.2. Note that, conditional on $Y_{n+1} = y$, the miscoverage event $Y_{n+1} \notin \hat{\mathcal{C}}_\alpha(X_{n+1})$ occurs if and only if $\hat{u}_y(X_{n+1}) \leq \alpha$, where $\hat{u}_y(X_{n+1})$ is defined as in (6). Therefore, it suffices to show $\mathbb{P}\left[\hat{u}_y(X_{n+1}) \leq \alpha \mid Y_{n+1} = y\right] \leq \alpha$ for any $\alpha \in (0,1)$. However, this is directly implied by Theorem 2.1, because the $\hat{u}_y(X_{n+1})$ calculated by Algorithm A5 is equivalent to the conformal p-value $\hat{u}_0(Z_{n+1})$ given by Algorithm A4 applied to the subset of the data in $\mathcal{D}_{\text{es-cal}}$ with $Y_i = y$, with the understanding that $Z_i = (X_i, Y_i)$ for all $i \in \{n+1\} \cup \mathcal{D}_{\text{es-cal}}$. □

Proof of Theorem A1. Note that $Y_{n+1} \notin \hat{\mathcal{C}}^{\text{m}}_\alpha(X_{n+1})$ if and only if $\hat{u}^{\text{marg}}(X_{n+1}; Y_{n+1}) \leq \alpha$, where $\hat{u}^{\text{marg}}(X_{n+1}; Y_{n+1})$ is defined as in (A17). Hence it suffices to show that $\mathbb{P}\left[\hat{u}^{\text{marg}}(X_{n+1}; Y_{n+1}) \leq \alpha\right] \leq \alpha$ for any $\alpha \in (0,1)$. This can be established using the same approach as in the proof of Theorem 2.1, setting $Z_i = (X_i, Y_i)$ for all $i \in \{n+1\} \cup \mathcal{D}_{\text{es-cal}}$. In fact, the maximum number of epochs $t^{\max}$ is fixed, the sequence of saved models is invariant to permutations of $\{n+1\} \cup \mathcal{D}_{\text{es-cal}}$, and the model $\hat{M}_{\text{ces}}$ selected according to (5) is also invariant. Thus, it follows that the nonconformity scores $\hat{S}_i$ are exchangeable with one another for all $i \in \{n+1\} \cup \mathcal{D}_{\text{es-cal}}$. □

Proof of Theorem 2.3. Consider an imaginary oracle algorithm producing an interval $\hat{C}^{\text{oracle}}_\alpha(X_{n+1})$ defined as $\hat{C}^{\text{oracle}}_\alpha(X_{n+1}) = \mathcal{B}_{l^*(Y_{n+1})} \bigcap \hat{C}_\alpha(X_{n+1}, \mathcal{B}_{l^*(Y_{n+1})})$, where $l^*(Y_{n+1})$ is the exact index of the bin $\mathcal{B}_l$ to which the true Y_{n+1} belongs. Clearly, this oracle is just a theoretical tool, not a practical method because the outcome value for the test point is unknown. However, this oracle is useful because it is easier to analyze, and it suffices to establish that $\mathbb{P}[Y_{n+1} \in \hat{C}^{\text{oracle}}_\alpha(X_{n+1})] \geq 1-\alpha$, for any $\alpha \in (0,1)$, since $\hat{C}_\alpha(X_{n+1}) \supseteq \hat{C}^{\text{oracle}}_\alpha(X_{n+1})$ almost-surely. The coverage property for the oracle can be established using an approach similar to that of the proof of Theorem 2.1, setting $Z_i = (X_i, Y_i)$ for all $i \in \{n+1\} \cup \mathcal{D}_{\text{es-cal}}$. In fact, the maximum number of epochs $t^{\max}$ is fixed, the sequence of saved models is invariant to permutations of $\{n+1\} \cup \mathcal{D}_{\text{es-cal}}$, and the model $\hat{M}_{\text{ces}}$ selected by the oracle according to (10) is also invariant. Thus, it follows that the oracle nonconformity scores $\hat{S}^*_i = \hat{S}_i(X_{n+1}, \mathcal{B}_{l^*(Y_{n+1})})$ are exchangeable with one another for all $i \in \{n+1\} \cup \mathcal{D}_{\text{es-cal}}$. Further, by construction of the prediction intervals (12), we know that the miscoverage event $Y_{n+1} \notin \hat{C}^{\text{oracle}}_\alpha(X_{n+1})$ occurs if and only if $\hat{S}^*_i > \hat{Q}^*_{1-\alpha}$, where $\hat{Q}^*_{1-\alpha}$ is the $\lceil (1-\alpha)(1+|\mathcal{D}_{\text{es-cal}}|) \rceil$-th largest value among all nonconformity scores $\hat{S}_i(X_{n+1}, \mathcal{B}_l)$. However, it is a well-known exchangeability result that $\mathbb{P}[\hat{S}^*_i \leq \hat{Q}^*_{1-\alpha}] \geq 1-\alpha$; see for example Lemma 1 in Romano et al. (2019). □

Proof of Theorem A2. Same as the proof of Theorem 2.3. □

Proof of Corollary A1. This corollary follows immediately from Theorem 2.3 because the prediction interval given by Algorithm A10 is always contained in that output by Algorithm A6. □

Proof of Corollary A3. Same as the proof of Corollary A1. □

Proof of Proposition A1. Note that $\hat{u}_0^{\text{naive}}(Z_{n+1}) = \hat{u}_0^{\text{naive}}(Z_{n+1}; t^*)$, hence

$$\begin{aligned}
\mathbb{P}\left[\hat{u}_0^{\text{naive}}(Z_{n+1}) > \alpha\right] &= \mathbb{E}\left[\mathbb{P}\left[\hat{u}_0^{\text{naive}}(Z_{n+1}; t^*) > \alpha \mid \mathcal{D}_{\text{es-cal}}\right]\right] \\
&\geq \mathbb{E}\left[\min_{t\in[T]} W_t\right] \\
&\geq \sup_{a\in[0,1]} a \cdot \mathbb{P}\left[\min_{t\in[T]} W_t \geq a\right] \\
&= \sup_{a\in[0,1]} a\left(1 - \mathbb{P}\left[\min_{t\in[T]} W_t \leq a\right]\right) \\
&\geq \sup_{a\in[0,1]} a\left(1 - T \cdot \mathbb{P}\left[W_t \leq a\right]\right),
\end{aligned}$$

where the last inequality follows from a union bound. To simplify the right-hand-side term above, let $a = I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} + 1 - l, l\right)$, where b is any large constant. Hence we obtain

$$\mathbb{P}\left[\hat{u}_0^{\text{naive}}(Z_{n+1}) > \alpha\right] \geq I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} + 1 - l, l\right) \cdot (1 - 1/b).$$

□

Proof of Corollary A3. Note that $Y_{n+1} \in \hat{C}_\alpha^{\text{naive}}(X_{n+1})$ if and only if $\hat{u}_{Y_{n+1}}^{\text{naive}}(X_{n+1}; t^*) > \alpha$. Let W_t denote the calibration conditional coverage $\mathbb{P}\left[\hat{u}_{Y_{n+1}}^{\text{naive}}(X_{n+1}; t) > \alpha \mid \mathcal{D}_{\text{es-cal}}\right]$. Then, we have

$$\mathbb{P}\left[Y_{n+1} \in \hat{C}_\alpha^{\text{naive}}(X_{n+1})\right] = \mathbb{E}\left[\mathbb{P}\left[\hat{u}_{Y_{n+1}}^{\text{naive}}(X_{n+1}; t^*) > \alpha \mid \mathcal{D}_{\text{es-cal}}\right]\right] = \mathbb{E}\left[W_{t^*}\right] \geq \mathbb{E}\left[\min_{t\in[T]} W_t\right].$$

The rest of the proof follows the same argument as in the proof of Proposition A1. □

Proof of Corollary A4. Let $\hat{S}_i(X_{n+1}, t) = |Y_i - \hat{\mu}_t(X_i)|$ denote the residual score calculated with model $t \in [T]$, for all $i \in \mathcal{D}_{\text{es-cal}}$. Note that $Y_{n+1} \in \hat{C}_\alpha^{\text{naive}}(X_{n+1})$ if and only if $\hat{S}_{X_{n+1}}(X_{n+1}, t^*) \leq \hat{Q}_{1-\alpha}$. Then, we just need to bound $W_t = \mathbb{P}\left[\hat{S}_{X_{n+1}}(X_{n+1}, t^*) \leq \hat{Q}_{1-\alpha} \mid \mathcal{D}_{\text{es-cal}}\right]$, and the rest of the proof follows the same steps as the proof of Proposition A1. □

Proof of Lemma A2. Recall that $l = \lfloor \alpha(n_{\text{es-cal}} + 1) \rfloor$, and define the following helpful notations:

$$\text{Beta}\left(n_{\text{es-cal}} + 1 - l, l\right) \coloneqq \text{Beta}\left(n_{\text{es-cal}} \cdot c, n_{\text{es-cal}} \cdot d\right), \quad \text{where } c = \frac{n_{\text{es-cal}} + 1 - l}{n_{\text{es-cal}}}, \quad d = \frac{l}{n_{\text{es-cal}}}.$$

Denote $\text{Gamma}(k, \theta)$ as the gamma distribution with shape parameter k and scale parameter θ. It is a well known fact that the beta distribution can be expressed as a ratio of gamma distributions as:

$$\text{Beta}\left(n_{\text{es-cal}} \cdot c, n_{\text{es-cal}} \cdot d\right) = \frac{\text{Gamma}(n_{\text{es-cal}} \cdot c, 1)}{\text{Gamma}(n_{\text{es-cal}} \cdot c, 1) + \text{Gamma}(n_{\text{es-cal}} \cdot d, 1)}.$$

Further, $\text{Gamma}(n_{\text{es-cal}} \cdot c, 1)$ can be seen as the distribution of a sum of $n_{\text{es-cal}} \cdot c$ independent exponentially distributed random variables with mean equal to 1; therefore, by the central limit theorem, $\text{Gamma}(n_{\text{es-cal}} \cdot c, 1)$ has an asymptotic Gaussian distribution as $n_{\text{es-cal}} \to \infty$. Denote $\Phi(x, \mu, \sigma^2)$ as the cumulative distribution function of a Gaussian random variable with mean μ and variance σ^2. Applying the delta method, it follows that, in the limit of large $n_{\text{es-cal}}$,

$$I\left(x; n_{\text{es-cal}} \cdot c, n_{\text{es-cal}} \cdot d\right) = \Phi\left(x; \frac{c}{c+d}, \frac{1}{n_{\text{es-cal}}} \cdot \frac{cd}{(c+d)^3}\right) + O\left(\frac{1}{n_{\text{es-cal}}}\right), \quad \text{for any } x \in [0, 1].$$

Since I and Φ are continuous and strictly increasing over $[0, 1]$, letting Φ^{-1} be the inverse Gaussian CDF, we have

$$\begin{aligned}
I^{-1}\left(\frac{1}{bT}; n_{\text{es-cal}} \cdot c, n_{\text{es-cal}} \cdot d\right) &= \Phi^{-1}\left(\frac{1}{bT}; \frac{c}{c+d}, \frac{1}{n_{\text{es-cal}}} \cdot \frac{cd}{(c+d)^3}\right) + O\left(\frac{1}{n_{\text{es-cal}}}\right) \\
&= \Phi^{-1}\left(\frac{1}{bT}; 1-\alpha, \frac{\alpha(1-\alpha)}{n_{\text{es-cal}}+1}\right) + O\left(\frac{1}{n_{\text{es-cal}}}\right) \\
&= (1-\alpha) + \sqrt{\frac{\alpha(1-\alpha)}{n_{\text{es-cal}}+1}} \cdot \Phi^{-1}\left(\frac{1}{bT}; 0, 1\right) + O\left(\frac{1}{n_{\text{es-cal}}}\right) \\
&= (1-\alpha) - \sqrt{\frac{\alpha(1-\alpha)}{n_{\text{es-cal}}+1}} \cdot \sqrt{2\log(bT)} + O\left(\frac{1}{\sqrt{n_{\text{es-cal}}\log(T)}}\right),
\end{aligned}$$

where the second equality is obtained by substituting c and d with their defined values and the last inequality follows from Equation 26.2.23 in Abramowitz & Stegun (1964) for sufficiently large T. □