Conformalized Deep Splines for
Optimal and Efficient Prediction Sets

Nathaniel Diamant   **Ehsan Hajiramezanali**   **Tommaso Biancalani**   **Gabriele Scalia**
Biology Research | AI Development (BRAID), Genentech

Abstract

Uncertainty estimation is critical in high-stakes machine learning applications. One effective way to estimate uncertainty is conformal prediction, which can provide predictive inference with statistical coverage guarantees. We present a new conformal regression method, *Spline Prediction Intervals via Conformal Estimation* (SPICE), that estimates the conditional density using neural-network-parameterized splines. We prove universal approximation and optimality results for SPICE, which are empirically reflected by our experiments. SPICE is compatible with two different efficient-to-compute conformal scores, one designed for size-efficient marginal coverage (SPICE-ND) and the other for size-efficient conditional coverage (SPICE-HPD). Results on benchmark datasets demonstrate SPICE-ND models achieve the smallest average prediction set sizes, including average size reductions of nearly 50% for some datasets compared to the next best baseline. SPICE-HPD models achieve the best conditional coverage compared to baselines. The SPICE implementation is made available.

1   INTRODUCTION

Uncertainty estimation is an essential problem in machine learning, especially for domains with high-stakes decisions such as medicine (Begoli et al., 2019), drug discovery (Scalia et al., 2020), and self-driving (Abdar et al., 2021). This work focuses on predictive inference for scalar regression, where the goal is to build

a *prediction set* rather than output a single value, in which the true target value is likely to fall. This formulation is an effective way to account for predictive uncertainty.

Conformal prediction (Vovk et al., 2005) is one of the most successful frameworks to approach building predictive sets. It only requires exchangeability of the data, does not make distributional and model assumptions, and comes with non-asymptotic coverage guarantee.

In this work, we present a deep regression model specifically designed for conformal prediction techniques, resulting in small and efficient-to-compute prediction sets with statistical coverage guarantees. As we will show, our proposed method has several theoretical and practical advantages over existing conformal prediction techniques and results in sharper predictive sets and improved conditional coverage on a range of benchmark datasets.

1.1   Conformal Prediction Background

We present an overview of split conformal prediction (Lei et al., 2015) for completeness and to introduce notation. For a more comprehensive introduction, see Angelopoulos and Bates (2022). Let $\{(x_i, y_i)\}_{i=1}^N$ denote N samples drawn from an unknown joint distribution with density function $f_{X,Y}(x, y)$. In split conformal, we randomly split the data into a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}$ and calibration data $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=N_{\text{train}}+1}^{N_{\text{train}}+N_{\text{cal}}}$.

1.1.1   Conformal Scores

The first step is to train a model $\hat{f}(x)$ on $\mathcal{D}_{\text{train}}$. Given the model, we define a conformal score $\mathcal{S}(x, y)$, which is higher the more the model's prediction disagrees with y. In a regression setting, a common way to build a conformal score is to divide the absolute residual by some heuristic uncertainty score, $u(x)$, such as the ensemble variance (Angelopoulos and Bates, 2022, §2.3). In the case where $\hat{f}(x)$ models the conditional density

Table 1: Comparison of the properties of different conformal regression methods. DISJOINT indicates whether a method can output multiple prediction intervals. OPT. MARGINAL indicates whether a method has oracle-optimal marginal size (see Section 1.2 for more information). OPT. CONDITIONAL indicates asymptotically optimal conditional size. DETERMINISTIC indicates whether prediction set computation is deterministic or stochastic. COMPUTATION is the computational complexity of computing a prediction set. *CHR has asymptotically optimal conditional interval size only for unimodal conditional distributions.

MODEL	DISJOINT	OPT. MARGINAL	OPT. CONDITIONAL	DETERMINISTIC	COMPUTATION
CQR (Romano et al., 2019)	✘	✘	✘	✔	$\mathcal{O}(1)$
CHR (Sesia and Romano, 2021)	✘	✘	✔*	✘	$\mathcal{O}([\# \text{ bins} \approx 10^3] \cdot [\# \text{ grid points} = 1{,}000])$
PCP (Wang et al., 2023)	✔	?	?	✘	$\mathcal{O}([\# \text{ samples} \approx 50] \cdot [\text{sample complexity}])$
SPICE-ND [OURS]	✔	✔	✘	✔	$\mathcal{O}([\# \text{ knots} \approx 30])$
SPICE-HPD [OURS]	✔	✘	✔	✔	$\mathcal{O}(\# \text{ knots} \cdot [\log \text{ precision} \approx 15])$

of $Y \mid X$, one can use $\mathcal{S}(x, y) = -\hat{f}(y \mid x)$. No matter what model and conformal score, the conformal prediction sets will have statistical guarantees. However, the modeling and score choices are critical for the size and properties of the prediction sets. A poor choice of model or score will lead to large and uninformative prediction sets.

1.1.2 Conformal Prediction Sets

In conformal prediction, the user specifies a desired miscoverage rate, $\alpha \in [0, 1]$. $1 - \alpha$ is called the *nominal coverage*. Assuming $(x_{\text{test}}, y_{\text{test}})$ and $\mathcal{D}_{\text{cal}}$ are exchangeable, split conformal builds prediction sets $\mathcal{C}(x)$ such that:

$$\mathbb{P}[y_{\text{test}} \in \mathcal{C}(x_{\text{test}})] \geq 1 - \alpha, \tag{1}$$

where the probability is taken over all possible draws of $(x_{\text{test}}, y_{\text{test}})$ and $\mathcal{D}_{\text{cal}}$. This property is known as *marginal coverage*, since for any individual $(x_{\text{test}}, y_{\text{test}})$ pair we have no guarantee. The stronger *conditional coverage* would be if Equation 1 held for each specific test sample.

In order to build the prediction sets, we calculate a quantile of the conformal scores of the calibration set, which we denote $\mathcal{S}_{\text{cal}} = \{\mathcal{S}(x, y) \colon (x, y) \in \mathcal{D}_{\text{cal}}\}$, as follows:

$$\hat{q} = \text{QUANTILE}\left[\mathcal{S}_{\text{cal}}, \frac{\lceil (N_{\text{cal}} + 1)(1 - \alpha)\rceil)}{N_{\text{cal}}}\right]. \tag{2}$$

Equation 2 is a slightly adjusted version of the nominal coverage quantile of the conformal score. The prediction set for a test sample is then:

$$\mathcal{C}(x_{\text{test}}) = \{y \colon \mathcal{S}(x_{\text{test}}, y) \leq \hat{q}\}. \tag{3}$$

Intuitively, the prediction set contains all the possible values of y that align well, to a degree determined by $\hat{q}$, with the model's prediction. A more accurate model will have a smaller $\hat{q}$, reflecting the fact that the true ys better align with the model's predictions in the calibration set.

1.2 Conformal Regression Desiderata

We have seen that any conformal score will result in marginal coverage. Therefore, we have to compare conformal methods based on other important properties of the prediction sets.

One key property is the *flexibility* of $\mathcal{C}(x)$ in its ability to cover multiple intervals of possible values for y. Sesia and Romano (2021) argue that the prediction set should be a single interval for improved interpretability. While this may be beneficial in some contexts, outputting multiple prediction intervals can make the prediction set much smaller in many cases (Wang et al., 2023). One case where this may be important is if there is a hidden categorical variable that leads to a multimodal conditional distribution for y, in which case a single interval could have to span between multiple modes and thereby include low-density regions. Additionally, applications with inherently multimodal observations have been highlighted across many domains (Izbicki et al., 2022).

Another important design principle is the size of the prediction intervals, assuming a perfect conditional density model (i.e. $\hat{f}(y \mid x) = f(y \mid x)$). Intuitively, given an oracle model, a well-designed conformal score should result in the smallest possible prediction intervals (Lei and Wasserman, 2014). We refer to this property, without any constraints besides marginal coverage, as *optimal marginal size*. When the score also results in conditional coverage, we call the property *optimal conditional size*. Optimal marginal size is achieved by under-covering highly variable conditional examples, which means that both properties cannot be simultaneously achieved (Sadinle et al., 2019). In practice, optimal conditional coverage can lead to improved non-oracle conditional coverage, while resulting in larger average prediction set sizes. One example of a regression score with optimal conditional coverage is the High Predictive Density (HPD) score (Izbicki et al., 2022).

Nathaniel Diamant, Ehsan Hajiramezanali, Tommaso Biancalani, Gabriele Scalia
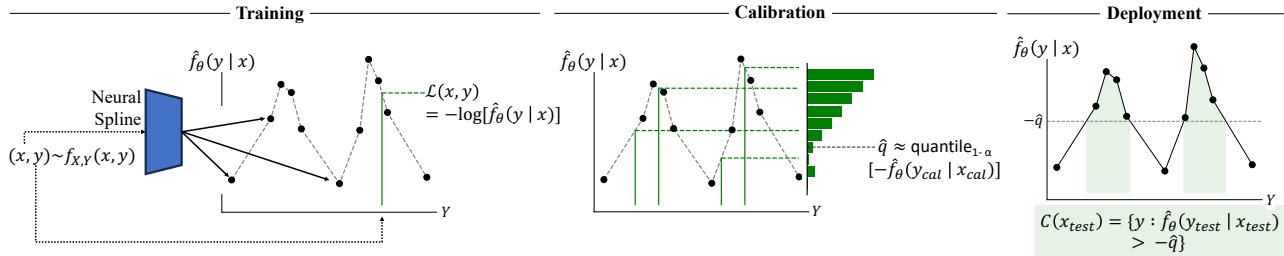
Figure 1: The three stages of SPICE-ND. SPICE-HPD shares the first stage but differs in the calibration and prediction stages.

Another practical consideration is the *complexity* of computing the conformal score and prediction sets. Some models, such as Conformalized Quantile Regression (CQR) (Romano et al., 2019), only require adding an offset to a prediction interval. Others require numeric integration (Izbicki et al., 2022), random sampling (Wang et al., 2023), or constructing large nested sets of intervals (Sesia and Romano, 2021). A high computational complexity can hinder applications in real-world settings. Additionally, some methods use randomized procedures to build the predictive set, resulting in stochastic predictions. In this regard, a *deterministic* computation could improve reproducibility, which is critical in many domains.

We summarize these desiderata in Table 1 and show that our method, *Spline Prediction Intervals via Conformal Estimation* (SPICE), achieves a uniquely strong combination of them.

Overall, we highlight the following contributions:

- We introduce a new and effective conformal regression method (SPICE), with two conformal score variants that can be efficiently computed on GPU, and release the code[1].
- SPICE prediction sets with the negative conditional density conformal score (SPICE-ND) have oracle-optimal marginal size.
- SPICE prediction sets with the HPD conformal score (SPICE-HPD) have asymptotically optimal conditional size.
- We prove that SPICE can uniformly approximate any continuous conditional density function and that it can express any finite union of prediction intervals (Section 2.2.3, 2.3.3).
- SPICE-ND models consistently achieve the smallest prediction sets compared to all baselines on standard conformal regression benchmarks.
- SPICE-HPD models achieve the best conditional coverage results compared to the baseline models.

[1] https://github.com/ndiamant/spice

2 Spline Prediction Intervals via Conformal Estimation (SPICE)

2.1 Overview

SPICE uses a neural network to build a spline estimate of the conditional density $f(Y \mid X)$. The conditional density estimate can then be used in either of two conformal scores to build prediction sets with different properties. Overall, SPICE consists of three main components:

1. A flexible neural network encoder which converts inputs into embeddings. This is standard across neural conformal-regression methods and enables regression on any modality a neural network can take as input.

2. A spline module which converts embeddings into conditional density functions (Section 2.2). This is the central innovation of SPICE. Low-order-splines as density functions enable efficient and closed-form computation of conformal scores prediction sets, while maintaining expressiveness.

3. One of two conformal scores (Section 2.3). One score results in optimal marginal size (Section 2.3.1), while the other satisfies optimal conditional size (Section 2.3.2). Both allow for expressive and flexible prediction sets.

See Figure 1 for a visual overview of SPICE.

2.2 Conditional Density Estimation

SPICE estimates the conditional density as fourth-degree-and-below polynomial splines, enabling integration and root-finding in closed form. The degree of a spline is the order of its polynomials, e.g. degree $n = 2$ for quadratic polynomials. For this work, we used first and second-degree splines, although the same approach can be applied to third and fourth-order polynomials, as they also have closed-form solutions.

We make the assumption that Y has support only on a finite interval, and in practice we zero-one normalize the data so that $0 \leq y < 1$. We note that we could map unbounded Y to zero-one using a function such as the sigmoid as noted by Barron and Sheu (1991), although the asymptotic interval size results would only apply with respect to the non-linearly transformed space. For generality and expressivity, SPICE differentiably maps covariates x to a spline in two stages. First, an encoder neural network $\text{NN}_{\theta_e}(x) = z$ encodes x. Next, two neural network modules map z to K sorted spline-knot positions and $n(K-1)+1$ knot heights, where n is the polynomial order. The differentiable parameterization of positions and heights is detailed in Appendix B.1.
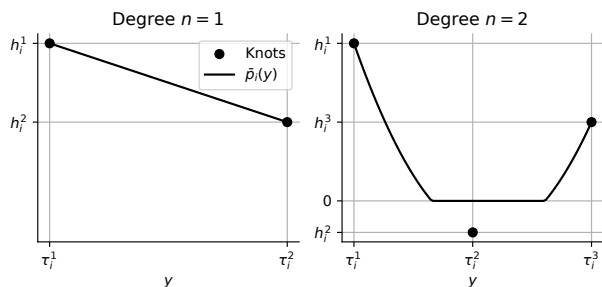

Figure 2: Zero-truncated Lagrange polynomials interpolating two (left) and three (right) points.

In this spline parameterization, there is a polynomial between every pair of consecutive knot positions (t_i, t_{i+1}). The positions and heights are used to construct $n+1$ points to uniquely specify each polynomial. We index the heights used to specify the i-th polynomial as h_i^j, where j ranges in $(1, \ldots, n+1)$. Note that some heights are redundantly indexed; i.e. $h_i^{n+1} = h_{i+1}^1$. In addition to the knots, we need $n-1$ evenly[2] spaced intermediate positions between t_i and t_{i+1} (Equation 4):

$$\tau_i^j = t_i + (j-1) \cdot \frac{(t_{i+1}-t_i)}{n} \text{ , where } j = 1, \ldots, n+1. \quad (4)$$

Using the positions and heights, we derive the unique order-n Lagrange polynomial, $p_i(y)$, that interpolates $\{(\tau_i^j, h_i^j)\}_{j=1}^{n+1}$. We constrain h_i^1 and h_i^{n+1} to be positive, but let all other h_i^j be any real number to maximize expressiveness. In order to ensure $p_i(y) > 0$, a requirement to build a density function, we define $\bar{p}_i(y) = \max(p_i(y), 0)$. Figure 2 illustrates examples for degrees one and two.

---

[2] For degrees three and four, Chebyshev points would likely be more effective (Trefethen, 2019).

2.2.1 Normalizing the Conditional Density

So far, we have shown how SPICE builds a non-negative spline from x. In order for the spline to be a valid density function, it needs to be normalized such that its integral evaluates to one. The normalizing constant is simply the sum of the integrals of the zero-truncated polynomials. The normalizing constant can be efficiently computed on GPU in $\mathcal{O}(Kn)$ time by integrating each polynomial $p_i(y)$ between its bounds and then subtracting the integrals between the polynomials' roots (Appendix B.2). The final normalized conditional density estimate is:

$$\hat{f}_\theta(y \mid x) = \bar{p}_\ell(y) \Big/ \sum_{i=1}^{K-1} \int_{t_i}^{t_{i+1}} \bar{p}_i(y')dy', \quad (5)$$

where ℓ is selected such that $t_\ell \leq y < t_{\ell+1}$. Note that ℓ can be found in $\mathcal{O}(\log K)$ time using binary search, since the knot positions are sorted, resulting in an overall complexity in $\mathcal{O}(Kn)$ for evaluating the estimated conditional density.

2.2.2 Training the Conditional Density Estimator

$\hat{f}_\theta(y \mid x)$ is differentiably parameterized with respect to the encoder, knot position, and knot height parameters ($\{\theta_e, \theta_t, \theta_h\}$). Therefore, SPICE can be trained using standard batch gradient descent methods to maximize the log-likelihood of the observed $(x, y) \in \mathcal{D}_{\text{train}}$.

2.2.3 Universal Approximation Theorem for Neural Splines

Both splines and neural networks are known to be very expressive, so it is not surprising that combining them results in a flexible function class. Without loss of generality to other intervals, we present the following universal approximation result on the unit cube.

Theorem 1. *Let $\mathbb{I}$ denote the unit interval $[0,1]$. Suppose for integer $d > 1$ that $f: (\mathbb{I}^{d-1} \times \mathbb{I}) \to \mathbb{R} \geq 0$ is continuous. Then for any $\epsilon > 0$, there exists a SPICE model without the unit-integral-normalization step, which we denote $\hat{f}(x,y): (\mathbb{I}^{d-1} \times \mathbb{I}) \to \mathbb{R} \geq 0$, for which $\sup_{(x,y) \in \mathbb{I}^d} |f(x,y) - \hat{f}(x,y)| < \epsilon$.*

Proof of Theorem 1 is provided in Appendix A.1.

Remark 1. *Theorem 1 may be extensible to non-continuous functions using recent results for approximating discontinuous functions (Ismailov, 2023).*

Corollary 1. *Unit-integral-normalized SPICE can approximate any continuous conditional density function f on $[0,1]^d$.*

Proof of Corollary 1 is provided in Appendix A.2.

2.3 Conformalizing SPICE

SPICE builds prediction intervals using either of two conformal scores: $\mathcal{S}_{\hat{f}}(x, y) = -\hat{f}_\theta(y \mid x)$ (Section 2.3.1) or the Highest Predictive Density (HPD) score, $\mathcal{S}_{\mathrm{HPD}}(x, y)$, defined in Izbicki et al. (2022) (Section 2.3.2). We refer to SPICE with the negative density score as SPICE-ND, and SPICE with the HPD-score as SPICE-HPD.

2.3.1 Negative Density Conformal Score

SPICE can use the negative estimated density of y, i.e. $\mathcal{S}_{\hat{f}}(x, y) = -\hat{f}_\theta(y \mid x)$, as an efficient and effective conformal score. As noted before, SPICE can compute the conditional density at y in $\mathcal{O}(Kn)$ time, which is much faster than the forward pass of the neural network encoder for practical values of K. A cutoff $\hat{q}$ with which to build prediction sets is found using the conformal scores computed on the calibration set as described in Section 1.1.1.

Given $\hat{q}$, the prediction set $\mathcal{C}_{\mathrm{ND}}(x)$ consists of all the values y such that $\hat{f}_\theta(y \mid x) > -\hat{q}$, following the conformal procedure described in Section 1.1.2. Figure 3 shows prediction sets for first and second-degree SPICE-ND. The prediction set $\mathcal{C}_{\mathrm{ND}}(x)$ can be computed in $\mathcal{O}(Kn)$ time since all it requires is finding the roots of low-order polynomials (Appendix B.3).

SPICE-ND achieves oracle-optimal marginal size due to the argument of Sadinle et al. (2019, §2.1) as a corollary of the Neyman–Pearson lemma. Under an argument similar to that of Izbicki et al. (2022), this could likely be extended into asymptotic optimality under consistency assumptions. However, we believe oracle-optimality is sufficient to justify the design choice of the negative density score. Indeed, we empirically found the negative density score yielded the smallest average prediction set sizes (Section 4.2). Interestingly, in a mixed frequentist-Bayesian setting and under some technical assumptions, using the negative-posterior-predictive-density of $y \mid x$ as the conformal score minimizes the Bayes-risk while preserving frequentist marginal coverage (Hoff, 2023).

2.3.2 HPD Conformal Score

Under the assumptions outlined for Izbicki et al. (2022, Theorem 25), which include the consistency of the conditional density estimator, the HPD-score results in asymptotically optimal conditional prediction set sizes, which is a stronger property than oracle-optimal conditional coverage. The HPD-score is defined as the negative[3] integral of the estimated conditional density

[3]For notational consistency we swapped the sign of the HPD-score from its original presentation.
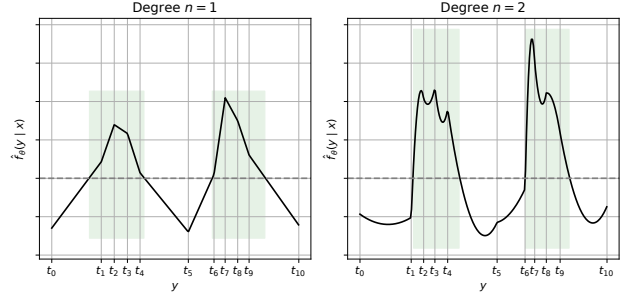

Figure 3: Examples of $\mathcal{C}_{\mathrm{ND}}(x)$ for first (left) and second degree (right) SPICE-ND. The horizontal dashed line is $-\hat{q}$, and the filled-in regions show the intervals of y in $\mathcal{C}_{\mathrm{ND}}(x)$.

where the density is less than the density of y:

$$\mathcal{S}_{\mathrm{HPD}}(x, y) = -\int_{y'\,:\,\hat{f}_\theta(y'|x)\leq\hat{f}_\theta(y|x)} \hat{f}_\theta(y' \mid x)dy'. \quad (6)$$

The HPD-score indicates a higher model error when y has lower density compared to other potential values of y. Notably, SPICE can compute the HPD-score in $\mathcal{O}(Kn)$ using the efficient root finding and integration properties of low-order polynomials.

In order to build prediction sets, a cutoff $\hat{q}$ is found using the split-conformal procedure described in Sections 1.1.1 resulting in the following prediction sets:

$$\mathcal{C}_{\mathrm{HPD}}(x) = \left\{ y: \int_{y'\,:\,\hat{f}_\theta(y'|x)\leq\hat{f}_\theta(y|x)} \hat{f}_\theta(y' \mid x)dy' > -\hat{q} \right\}. \quad (7)$$

Computing an HPD prediction set for SPICE is more involved than for the negative density score, and is done approximately to arbitrary precision using the bisection method (Appendix B.4). This procedure has an asymptotic runtime in $\mathcal{O}\left(Kn \log_2 \frac{1}{\epsilon}\right)$, where ϵ is the desired precision. In practice, we only need to run the bisection method for 24 steps to have higher precision than 32-bit floats, and all the computations can be done on GPU.

2.3.3 Prediction Sets Expressible by SPICE

Here, we examine which prediction sets SPICE is capable of expressing. It is clear that given control of the conformal cutoff $\hat{q}$ and enough knots, SPICE could express any finite union of intervals in $[0, 1]$. However, unique to SPICE is its flexibility to build any prediction set even without control of $\hat{q}$. For example, PCP (Wang et al., 2023) can express up to as many intervals as samples are taken from its conditional generative model (Section 3), but the smallest any of those intervals can be is $2\hat{q}$. That extra flexibility may partially explain the empirical reduction in interval sizes

achieved by SPICE (Section 4.2). We formalize this flexibility in the following theorem.

Theorem 2. *Let* $\mathscr{P} = \bigcup_{i=1}^{m}[a_i, b_i] \subseteq [0, 1]$ *where* $a_i < b_i < a_{i+1}$ *and* $1 \leq m \leq M$ *for some natural numbers* m *and* M. *Then the following are true with* $K = 4M + 2$:

1. *Given any* $\hat{q}$: $0 < -\hat{q} \cdot |\mathscr{P}| < 1$ *there exists a SPICE-ND model with* K *knots such that* $\mathcal{C}_{\mathrm{ND}}(x) = \mathscr{P}$ *for some arbitrary* x.

2. *Given any* $0 < -\hat{q} < 1$ *there exists a SPICE-HPD model with* K *knots such that* $\mathcal{C}_{\mathrm{HPD}}(x) = \mathscr{P}$ *for some arbitrary* x.

See Appendix A.3 for a proof. Theorem 2 shows that SPICE can express any union of intervals with a number of knots linear in the number of intervals given any conformal cutoff.

3 RELATED WORK

Durkan et al. (2019) introduced a monotonically increasing cubic spline, which served as an invertible layer in the context of normalizing flows. This was part of the inspiration for the spline conditional density function, and we leveraged their technique for differentiably parameterized spline bin-widths.

Conformalized Quantile Regression (CQR) (Romano et al., 2019) has become one of the most common conformal regression methods. CQR uses the pinball-loss (Steinwart and Christmann, 2011) to train a neural network to estimate the α and $1-\alpha$ quantiles of $Y \mid x$, where α is a tunable hyperparameter. The conformal procedure results in an offset $\hat{q}$ which leads to prediction set $\mathcal{C}(x) = [\hat{f}_{\alpha}(x) - \hat{q}, \hat{f}_{1-\alpha}(x) + \hat{q}]$. Asymptotically, CQR does have conditional coverage, but does not always find the minimal prediction interval (Sesia and Romano, 2021).

Conditional Histogram Regression (CHR) (Sesia and Romano, 2021) can be seen as a more flexible extension of CQR with the additional goal of finding the single minimal prediction interval with asymptotic conditional coverage. Rather than estimating just two quantiles, CHR estimates many (up to 1,000 in their experiments). The CHR conformal score involves building a series of nested intervals, each being a subset or superset of an initial interval following the nested sets paradigm (Gupta et al., 2022). Each interval is designed to be the minimal interval that is predicted to contain a target probability mass, and that satisfies nesting with the initial interval. The conformal score is the index of the nested interval that contains the true y. Assuming that the conditional distribution is unimodal and that the quantile prediction model is consistent, CHR asymptotically finds the smallest possible interval and achieves conditional coverage.

Probabilistic Conformal Regression (PCP) (Wang et al., 2023) is a method to adapt any conditional generative model to conformal regression. The PCP conformal score is calculated by sampling from the estimated conditional distribution (usually around 50 samples), and then finding the minimum Euclidean distance between y and any of the samples. The prediction set is then defined as the union of balls around samples from the target distribution, with the ball radius equal to the quantile computed from the calibration set. PCP thus enables disjoint prediction intervals, and therefore tends to find smaller prediction sets than CHR or CQR. However, PCP does not provide guarantees for asymptotic conditional coverage. Additionally, both conformal scores and predicted intervals rely on sampled points, making the estimation stochastic. In the following, we use the strongest PCP variant shown by Wang et al. (2023), HPD-PCP-MixD, which uses a Gaussian mixture for the conditional distribution and rejects the bottom 20% least likely samples from the score and prediction set calculations.

Distributional Conformal Prediction (DCP) introduced by Chernozhukov et al. (2021) is a conformal regression method which uses an estimated conditional cumulative distribution and has similar asymptotic results to CHR. In experiments, Sesia and Romano (2021) found CHR outperformed DCP, so we did not include DCP as a baseline. Dist-split and CD-split (Izbicki et al., 2020) are two conformal regression methods that work with conditional density estimators. These methods were built on by Izbicki et al. (2022) to design the HPD conformal score, which we leverage with SPICE-HPD to gain optimal conditional size. The HPD-score (Izbicki et al., 2022) was tested with the FlexCode conditional density estimator (Izbicki and Lee, 2017), which requires many hyperparameters to tune, and was not designed for neural network density estimation, so we did not include it as a baseline. Both Dist-split and CD-split were outperformed by PCP in Wang et al. (2023), so we also did not include them as baselines and used PCP as a stronger baseline.

4 EXPERIMENTS

We evaluated SPICE-ND, SPICE-HPD, and the baseline models on eight datasets commonly used for conformal regression benchmarking (dataset details in Appendix C.3). Each model had approximately the same number of parameters and the same encoder architecture to ensure differences in performance were not caused by architecture. Hyperparameters were se-

Table 2: Normalized marginal size results on benchmark datasets. The nominal coverage was set to 90%. **Bold** indicates the best mean; <u>underline</u> indicates the second best. MEAN is the average size across all datasets.

MODEL	BIKE	BIO	BLOG	MEPS19	MEPS20	MEPS21	STAR	TEMP.	MEAN
CQR	1.00 ± 0.01	0.99 ± 0.00	0.20 ± 0.01	0.98 ± 0.02	0.91 ± 0.02	0.87 ± 0.02	0.89 ± 0.01	0.89 ± 0.01	0.84 ± 0.09
CHR	0.89 ± 0.01	0.98 ± 0.00	0.14 ± 0.01	0.74 ± 0.01	0.70 ± 0.01	0.66 ± 0.01	0.91 ± 0.01	0.89 ± 0.01	0.74 ± 0.09
PCP	0.23 ± 0.00	0.59 ± 0.00	0.16 ± 0.01	0.81 ± 0.01	0.82 ± 0.02	0.79 ± 0.02	0.89 ± 0.01	<u>0.28 ± 0.00</u>	0.57 ± 0.11
Hist.	0.26 ± 0.00	**0.53 ± 0.00**	0.38 ± 0.00	0.66 ± 0.03	0.59 ± 0.07	0.77 ± 0.01	0.94 ± 0.01	0.31 ± 0.01	0.56 ± 0.08
SPICE-ND ($n=1$)	**0.20 ± 0.00**	0.59 ± 0.02	**0.09 ± 0.00**	0.57 ± 0.01	<u>0.55 ± 0.01</u>	<u>0.52 ± 0.01</u>	0.90 ± 0.01	**0.27 ± 0.00**	<u>0.46 ± 0.09</u>
SPICE-HPD ($n=1$)	0.24 ± 0.00	0.62 ± 0.02	0.15 ± 0.01	0.87 ± 0.01	0.82 ± 0.01	0.87 ± 0.02	0.93 ± 0.01	0.29 ± 0.00	0.60 ± 0.11
SPICE-ND ($n=2$)	<u>0.22 ± 0.00</u>	<u>0.55 ± 0.01</u>	<u>0.10 ± 0.00</u>	**0.32 ± 0.04**	**0.35 ± 0.03**	**0.37 ± 0.03**	**0.83 ± 0.01**	0.28 ± 0.00	**0.38 ± 0.08**
SPICE-HPD ($n=2$)	0.23 ± 0.00	0.58 ± 0.01	0.19 ± 0.01	<u>0.45 ± 0.04</u>	1.11 ± 0.20	0.64 ± 0.14	<u>0.84 ± 0.01</u>	0.28 ± 0.01	0.54 ± 0.11

Table 3: Approximate label-conditional coverage results on benchmark datasets at 90% nominal coverage. **Bold** indicates the best mean; <u>underline</u> indicates the second best. MEAN is the average across all datasets.

MODEL	BIKE	BIO	BLOG	MEPS19	MEPS20	MEPS21	STAR	TEMP.	MEAN
CQR	0.47 ± 0.01	0.52 ± 0.00	0.06 ± 0.01	0.52 ± 0.00	0.43 ± 0.01	0.41 ± 0.01	0.76 ± 0.01	0.69 ± 0.01	0.48 ± 0.07
CHR	0.47 ± 0.01	0.53 ± 0.00	0.08 ± 0.01	0.53 ± 0.00	0.46 ± 0.01	0.43 ± 0.01	0.73 ± 0.01	0.67 ± 0.01	0.49 ± 0.07
PCP	0.80 ± 0.01	0.84 ± 0.00	<u>0.37 ± 0.01</u>	0.53 ± 0.01	0.51 ± 0.01	0.50 ± 0.01	0.81 ± 0.01	0.87 ± 0.01	0.65 ± 0.07
Hist.	0.75 ± 0.01	**0.85 ± 0.00**	0.02 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.03 ± 0.01	0.81 ± 0.03	0.86 ± 0.00	0.42 ± 0.15
SPICE-ND ($n=1$)	0.83 ± 0.01	0.83 ± 0.01	0.24 ± 0.01	0.52 ± 0.01	0.50 ± 0.01	0.47 ± 0.01	<u>0.81 ± 0.01</u>	**0.87 ± 0.00**	0.63 ± 0.08
SPICE-HPD ($n=1$)	**0.86 ± 0.00**	0.83 ± 0.01	0.25 ± 0.01	**0.61 ± 0.00**	**0.60 ± 0.01**	**0.61 ± 0.01**	**0.83 ± 0.01**	<u>0.87 ± 0.00</u>	**0.68 ± 0.07**
SPICE-ND ($n=2$)	0.80 ± 0.01	0.84 ± 0.01	0.21 ± 0.01	0.56 ± 0.01	0.53 ± 0.01	0.49 ± 0.01	0.76 ± 0.01	0.87 ± 0.00	0.63 ± 0.08
SPICE-HPD ($n=2$)	<u>0.85 ± 0.01</u>	<u>0.84 ± 0.01</u>	**0.42 ± 0.01**	<u>0.57 ± 0.02</u>	<u>0.59 ± 0.02</u>	<u>0.54 ± 0.02</u>	0.77 ± 0.01	0.86 ± 0.01	<u>0.68 ± 0.06</u>

lected using grid search, with the same number of hyperparameter combinations searched over for each method. Further optimization details are in Appendix C.5. Results were computed on a heldout test dataset using 20 replicates with different random initializations and train-validation splits for each model (data split details in Appendix C.4).

4.1 Discretized y Baseline

We also introduce a simple baseline for conformal regression by discretizing the domain of y into equally sized bins, and using a conformal technique commonly used in the classification setting. Specifically, we used the negative model-predicted probability of the discretized y as the conformal score, which is the optimal choice in terms of expected prediction set size for an oracle classifier (Sadinle et al., 2019). To the best of our knowledge, other studies on conformal regression have not incorporated this baseline, although a similar approach has been used in other contexts for conditional density estimation (Chen et al., 2022). Given its simplicity, properties, and adequate performance, we believe that including it should become common practice in future works. This model is denoted as *Hist.* in the following results. Details are in Appendix C.7.

4.2 Marginal Size Results

All models achieved their marginal coverage targets as expected, so we compared the average interval sizes

on heldout data. We normalized the size measurement to the size achieved by a marginal histogram of the training y in order to make sizes more comparable across datasets (Appendix C.8). As shown in Table 2, SPICE-ND with $n = 1, 2$ consistently resulted in the smallest interval sizes. SPICE-HPD performed worse in terms of average interval size, but was competitive with the baseline methods. Of the baseline methods, PCP and our histogram baseline were the most performant. Overall, SPICE-ND ($n = 2$) had the smallest intervals, perhaps due to the increased flexibility of quadratic versus linear interpolation.

4.3 Conditional Coverage Results

Conditional coverage is the coverage of a prediction set for a specific set of covariates, x. It is inaccessible except for synthetic data, so it can only be approximated for real data. Following the recommendation of Angelopoulos and Bates (2022), we measured *label-conditional coverage* as our conditional coverage metric (details in Appendix C.9). For completeness, we also computed the worst-slab coverage (WSC) (Cauchois et al., 2021; Romano et al., 2020) to approximate conditional coverage (Appendix D.1).

As shown in Table 3, SPICE-HPD with $n = 1$ and $n = 2$ had the best and second best conditional coverage respectively on average across all datasets. In six out of eight datasets, SPICE-HPD outperformed SPICE-ND and otherwise had close to the same conditional coverage, reflecting the benefit of the optimal
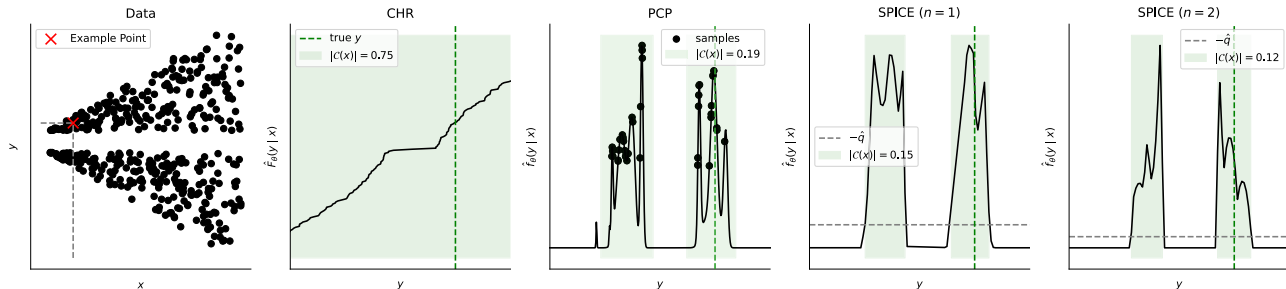
Figure 4: Example prediction sets for SPICE-ND and two baselines on a synthetic data sample.
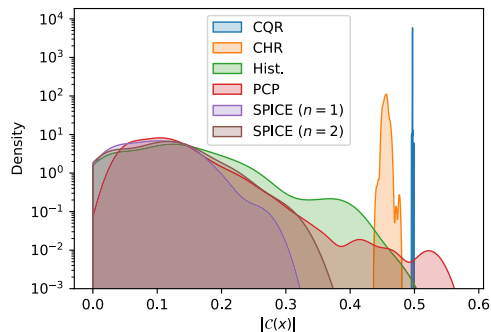
conditional size property.

Figure 5: Kernel-density plots of the sizes of predictive sets for each model on the BIKE dataset.

4.4 Analysis of Interval Sizes

In order to understand the smaller predictive set sizes of SPICE, we analyzed the distributions of the set sizes for each model on the BIKE dataset (Figure 5). The CQR and CHR set sizes were relatively invariant across test samples compared to the methods with more flexible prediction sets. PCP had much more varied sizes, potentially reflecting the stochasticity of its predictive set construction procedure. PCP was also less likely to express the smallest intervals, which may be due to the restriction of the smallest possible interval size it can express (Section 2.3.3).

These differences were supported by results on synthetic data with heteroscedastic bimodal conditional distributions (described in Appendix C.10). We show example prediction sets in Figure 4. We found that CHR and CQR were forced to cover low-density regions due to their single-interval prediction set constraint. PCP prediction intervals captured the bimodality, but also covered some low predictive density regions. This is due to the $\hat{q}$ radius around conditional samples extending outside of the high-density regions. In contrast, the SPICE-ND prediction sets covered only high-density regions.

5 DISCUSSION

SPICE-ND consistently achieved the smallest average prediction set sizes across experiments, and SPICE-HPD the best conditional coverage results, reflecting their marginal and conditional size design goals. Nonetheless, SPICE-ND also performed well in terms of conditional coverage. This suggests that SPICE-ND may be a suitable general-purpose choice when marginal coverage is key and conditional coverage is not critical, due to its computational efficiency (Table 1). Degree-two spline SPICE models resulted in smaller interval sizes and comparable conditional coverage results. The flexibility of quadratic compared to linear functions may explain this result, and suggests cubic splines could be a valuable avenue for future exploration.

The strong performance of Hist., the discrete classifier baseline (Section 4.1), highlights the importance of incorporating competitive baselines when evaluating new conformal methods, and should be considered in future conformal regression studies.

A limitation of SPICE is its inability to directly handle multi-target regression problems. An interesting area of future work is extending SPICE to the multi-target case, perhaps by modeling the joint density. For multi-target problems, we currently recommend PCP, despite its lack of optimality results, because of its flexibility of choice in conditional generative models and simple prediction set computation.

6 CONCLUSION

We introduced SPICE, which provides an expressive, efficient, and practical method for uncertainty estimation in scalar regression problems. SPICE achieves a unique combination of advantages and consistently outperforms existing methods, resulting in sharper predictive sets and improved conditional coverage. The effectiveness of SPICE highlights the benefits of co-designing the regression model and conformal

scores. The code release will support adoption in applications such as medicine and drug discovery, where reliable uncertainty estimation is critical. An important direction for future work is applying SPICE to selection problems, such as portfolio optimization and experimental design (Jin and Candès, 2023; Stanton et al., 2023). Among many potential application areas, survival analysis (Wang et al., 2019) represents a promising use case of SPICE, as conditional coverage is particularly valuable in this setting, and efficient evaluation of the conditional CDF could enable survival predictions with statistical coverage guarantees. Finally, a promising research direction involves the addition of SPICE uncertainty quantification to foundational models, which are becoming a critical tool across many fields, including biology and drug discovery (Heimberg et al., 2023; Theodoris et al., 2023). Improved uncertainty estimation could enhance their reliability and applicability, ultimately improving their effectiveness in discovery problems.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

C.M. Achilles, Helen Pate Bain, Fred Bellott, Jayne Boyd-Zaharias, Jeremy Finn, John Folger, John Johnston, and Elizabeth Word. Tennessee's Student Teacher Achievement Ratio (STAR) project, 2008.

Anastasios N. Angelopoulos and Stephen Bates. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification, December 2022. arXiv:2107.07511.

Andrew R. Barron and Chyong-Hwa Sheu. Approximation of density functions by sequences of exponential families. *Annals of Statistics*, 19:1347–1369, 1991.

Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1(1):20–23, 2019.

Krisztian Buza. BlogFeedback. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C58S3F.

Maxime Cauchois, Suyash Gupta, and John C Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction.

The Journal of Machine Learning Research, 22(1): 3681–3722, 2021.

Bing Chen, Mazharul Islam, Jisuo Gao, and Lin Wang. Deconvolutional density network: Modeling free-form conditional distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6183–6192, 2022.

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

Dongjin Cho, Cheolhee Yoo, Jungho Im, and Dong-Hyun Cha. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas. *Earth and Space Science*, 7(4), 2020.

Joel W Cohen, Steven B Cohen, and Jessica S Banthin. The medical expenditure panel survey: a national information resource to support healthcare cost research and inform policy and practice. *Medical care*, pages S44–S50, 2009.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Cubic-Spline Flows, 2019. arXiv:1906.02145.

Hadi Fanaee-T. Bike Sharing Dataset. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

Chirag Gupta, Arun K. Kuchibhotla, and Aaditya Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, 127:108496, 2022.

Graham Heimberg, Tony Kuo, Daryle DePianto, Tobias Heigl, Nathaniel Diamant, Omar Salem, Gabriele Scalia, Tommaso Biancalani, Jason Rock, Shannon Turley, Héctor Corrada Bravo, Josh Kaminker, Jason A. Vander Heiden, and Aviv Regev. Scalable querying of human cell atlases via a foundational model reveals commonalities across fibrosis-associated macrophages. *bioRxiv*, 2023.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv*, 2016. arXiv:1606.08415.

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

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

Vugar E. Ismailov. A three layer neural network can represent any multivariate function. *Journal of Mathematical Analysis and Applications*, 523(1): 127096, 2023.

Rafael Izbicki and Ann B. Lee. Converting high-dimensional regression to high-dimensional conditional density estimation. *Electronic Journal of Statistics*, 11(2):2800–2831, 2017.

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

Rafael Izbicki, Gilson Shimizu, and Rafael B. Stern. CD-split and HPD-split: Efficient Conformal Regions in High Dimensions. *Journal of Machine Learning Research*, 23(87):1–32, 2022.

Ying Jin and Emmanuel J Candès. Selection by prediction with conformal p-values. *Journal of Machine Learning Research*, 24(244):1–41, 2023.

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

Jing Lei, Alessandro Rinaldo, and Larry Wasserman. A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 74(1):29–43, 2015.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

M. J. D. Powell. *Approximation Theory and Methods*, chapter 20-21. Cambridge University Press, 1981.

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

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized Quantile Regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234, 2019.

Gabriele Scalia, Colin A. Grambow, Barbara Pernici, Yi-Pei Li, and William H. Green. Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction. *Journal of Chemical Information and Modeling*, 60(6):2697–2717, 2020.

Matteo Sesia and Yaniv Romano. Conformal Prediction using Conditional Histograms. In *Advances in Neural Information Processing Systems*, volume 34, pages 6304–6315, 2021.

Samuel Stanton, Wesley Maddox, and Andrew Gordon Wilson. Bayesian optimization with conformal prediction sets. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 959–986. PMLR, 2023.

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

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, pages 1–9, 2023.

Lloyd N Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*, chapter 2. SIAM, 2019.

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

Ping Wang, Yan Li, and Chandan K Reddy. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

Zhendong Wang, Ruijiang Gao, Mingzhang Yin, Mingyuan Zhou, and David Blei. Probabilistic Conformal Prediction Using Conditional Random Samples. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 8814–8836. PMLR, 2023.

Checklist

1. For all models and algorithms presented, check if you include:

Nathaniel Diamant, Ehsan Hajiramezanali, Tommaso Biancalani, Gabriele Scalia

(a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
When technical assumptions are necessary, they are noted (e.g. continuity for Theorem 1). The SPICE model and predictions set computations are thoroughly described in the main text and the supplement.

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
We included a zipped directory including all the source code including python package requirements and installation instructions as supplementary material.

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [Yes]

(b) Complete proofs of all theoretical results. [Yes]
Proofs are provided in the supplement.

(c) Clear explanations of any assumptions. [Yes]
Assumptions are part of each proof.

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
All code to reproduce the results is at `https://github.com/ndiamant/spice`.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
These are described in Appendices C.6, C.5, and C.4.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
All errors were computed as the standard error of the mean over 20 random seeds.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
All experiments were run on an internal cluster of A100 NVIDIA GPUs.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]
Dataset sources are in Appendix C.3.

(b) The license information of the assets, if applicable. [Yes]
License information is included in Appendix C.3.

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A PROOFS

A.1 Proof of Universal Approximation Theorem

We reproduce Theorem 1 here for easy reference.

Theorem. *Let $\mathbb{I}$ denote the unit interval $[0, 1]$. Suppose for integer $d > 1$ that $f \colon (\mathbb{I}^{d-1} \times \mathbb{I}) \to \mathbb{R} \geq 0$ is continuous. Then for any $\epsilon > 0$, there exists a SPICE model without the normalization to unit-integral step, which we denote $\hat{f}(x, y) \colon (\mathbb{I}^{d-1} \times \mathbb{I}) \to \mathbb{R} \geq 0$, for which $\sup_{(x,y) \in \mathbb{I}^d} |f(x, y) - \hat{f}(x, y)| < \epsilon$.*

This theorem is general to single-hidden-layer neural network encoders for the spline-knot positions and heights. It also holds for our specific parameterization of spline knots (Appendix B.1), since the softmax is a continuous, surjective function onto $(0, 1)$.

Proof. Let $\phi \in \mathbb{R}^K$ be a vector of spline parameters for natural number K and let $s(\phi, y)$ for $y \in \mathbb{I}$ be the resulting spline function. Let $\mathscr{S}(x) \colon \mathbb{I}^{d-1} \to \mathbb{R}^K$ be a function mapping x to spline parameters such that for all $x \in \mathbb{I}^{d-1}$

$$\sup_{y \in \mathbb{I}} |f(x, y) - s(\mathscr{S}(x), y)| < \epsilon/3. \tag{8}$$

We know $\mathscr{S}(x)$ exists for sufficiently large K by the fact that splines can uniformly approximate continuous functions (Powell, 1981, Theorem 20.11).

Next, we define function $E \colon (\mathbb{I}^{d-1} \times \mathbb{R}^K) \to \mathbb{R} \geq 0$ as

$$E(x, \phi) = \sup_{y \in \mathbb{I}} |s(\mathscr{S}(x), y) - s(\phi, y)|. \tag{9}$$

Since splines are continuous in their knot positions and heights, the function E is also continuous in ϕ by the continuity of compositions of continuous functions. Therefore, there exists a δ such that for all $x \in \mathbb{I}^{d-1}$ and functions $\mathscr{S}'(x) \colon \mathbb{I}^{d-1} \to \mathbb{R}^K$ such that $||\mathscr{S}'(x) - \mathscr{S}(x)||_\infty < \delta$ we have that

$$E\left(x, \mathscr{S}'(x)\right) = \sup_{y \in \mathbb{I}} |s\left(\mathscr{S}(x), y\right) - s\left(\mathscr{S}'(x), y\right)| < \epsilon/3. \tag{10}$$

We now use a classic universal approximation theorem for neural networks to show that a neural network exists that can be $\mathscr{S}'(x)$. Hornik (1991) showed that a single-hidden-layer feed-forward neural network can uniformly approximate any continuous function on a compact subset of $\mathbb{R}$, such as $\mathbb{I}^{d-1}$. Therefore, there exists feed-forward neural network $\mathscr{N} \colon \mathbb{I}^{d-1} \to \mathbb{R}^K$ such that

$$\sup_{x \in \mathbb{I}^{d-1}} ||\mathscr{N}(x) - \mathscr{S}(x)||_\infty < \delta. \tag{11}$$

Finally, by combining Equations 8 and 11, we have by the triangle inequality that

$$\sup_{(x,y) \in \mathbb{I}^d} |s(\mathscr{N}(x), y) - f(x, y)| \leq \sup_{(x,y) \in \mathbb{I}^d} |f(x, y) - s(\mathscr{S}(x), y)| + \sup_{(x,y) \in \mathbb{I}^d} |s(\mathscr{N}(x), y) - s(\mathscr{S}(x), y)|$$

$$\leq 2\epsilon/3 < \epsilon.$$

We have shown that a feed-forward network which outputs a spline can uniformly approximate any continuous function on the hypercube. This completes the proof.

$\square$

A.2   Proof of Universal Approximation of Conditional Density Functions

We reproduce Corollary 1 here for convenience.

Corollary. *Unit-integral-normalized SPICE can approximate any continuous conditional density function f on $[0,1]^d$.*

Proof. Let $\hat{f}$ be an unnormalized approximation of f such that $|f(x,y) - \hat{f}(x,y)| < \delta$ for some $\delta > 0$. By Theorem 1, we know such $\hat{f}$ exists. For any x, we define normalized SPICE as:

$$\tilde{f}(x,y) = \frac{\hat{f}(x,y)}{\int_0^1 \hat{f}(x,y)dy}.$$

Note that the normalization constant is bounded on both sides: $1 - \delta < \int_0^1 \hat{f}(x,y)dy < 1 + \delta$. Using this fact we can bound the approximation error of $\tilde{f}$.

$$\sup_{(x,y)\in\mathbb{I}^d} |f(x,y) - \tilde{f}(x,y)| \leq \max \left\{ \sup_{(x,y)\in\mathbb{I}^d} \left| f(x,y) - \frac{\hat{f}(x,y)}{1-\delta} \right|, \sup_{(x,y)\in\mathbb{I}^d} \left| f(x,y) - \frac{\hat{f}(x,y)}{1+\delta} \right| \right\}$$

$$\leq \max \left\{ \sup_{(x,y)\in\mathbb{I}^d} \left| f(x,y) - \frac{f(x,y)+\delta}{1-\delta} \right|, \sup_{(x,y)\in\mathbb{I}^d} \left| f(x,y) - \frac{f(x,y)-\delta}{1+\delta} \right| \right\}.$$

Let M be the maximum absolute value of f, which exists since f is continuous on a closed interval. Then by setting $\delta < 1/2$ and rearranging, we get:

$$\sup_{(x,y)\in\mathbb{I}^d} |f(x,y) - \tilde{f}(x,y)| \leq \max \left\{ \delta 2|M-1|, \delta\frac{2}{3}|M-1| \right\}.$$

By setting $\delta < \min \{\epsilon/(2|M-1|), 1/2\}$ we attain our result.

$\square$

A.3   Proof of SPICE Interval Expressiveness

Theorem. *Let $\mathscr{P} = \bigcup_{i=1}^m [a_i, b_i] \subseteq [0,1]$ where $a_i < b_i < a_{i+1}$ and $1 \leq m \leq M$ for some natural numbers m and M. Then the following are true with $K = 4M + 2$:*

1. *Given any $\hat{q}: 0 < -\hat{q} \cdot |\mathscr{P}| < 1$ there exists a SPICE-ND model with K knots such that $\mathcal{C}_{\mathrm{ND}}(x) = \mathscr{P}$ for some arbitrary x.*

2. *Given any $0 < -\hat{q} < 1$ there exists a SPICE-HPD model with K knots such that $\mathcal{C}_{\mathrm{HPD}}(x) = \mathscr{P}$ for some arbitrary x.*

The proof of Theorem A.3 is a constructive proof and focuses on degree-one splines, since any higher degree spline can express a lower degree one. The construction is not necessarily the most efficient way to build our target prediction set in terms of number of knots, but it shows that any such prediction set can be constructed with a linear number of knots. See Figure 6 for a visualization of the constructed prediction sets.

Proof. We begin by noting that, as a result of Theorem 1, SPICE can express any combination of knot positions $\{t_i\}_{i=1}^K$ and knot heights $\{h_i\}_{i=1}^K$ such that $\hat{f}_\theta$ integrates to one. Therefore, we will show that there exists a combination of knot parameters for any given $\hat{q}$ that results in the goal prediction set $\mathscr{P}$ and that results in $\hat{f}_\theta$ integrating to one.
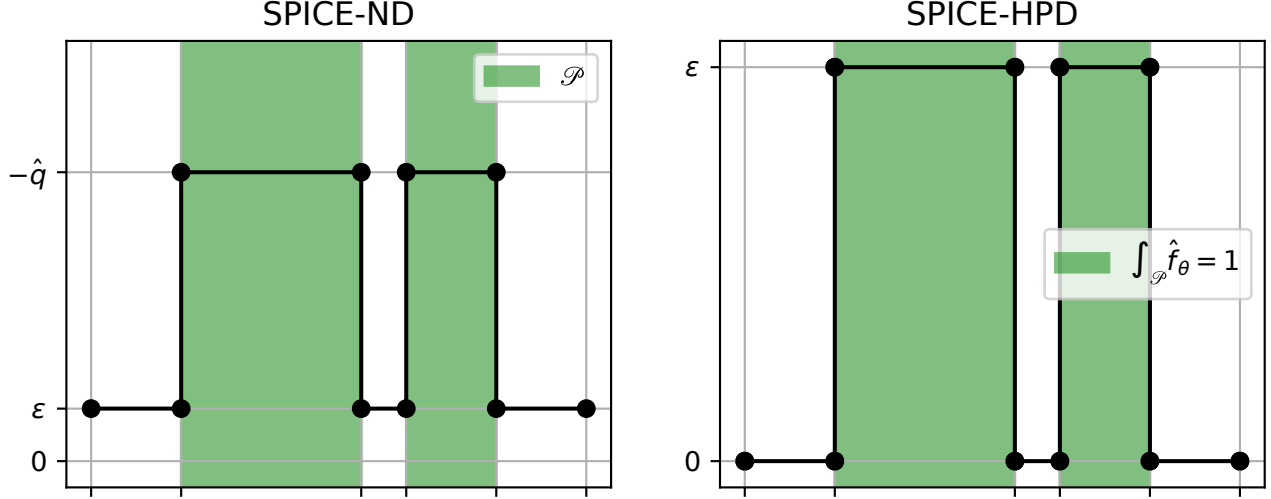
Figure 6: Method for constructing an arbitrary union of intervals $\mathscr{P}$ as prediction sets by SPICE-ND and SPICE-HPD.

The SPICE-ND case. For SPICE-ND, the prediction set is $\mathcal{C}_{\mathrm{ND}}(x) = \{y \colon \hat{f}_\theta(y \mid x) > -\hat{q}\}$. We start by constructing the first interval $[a_1, b_1]$ using three knots. For convenience, we will define $c = -\hat{q}$. Set $(t_1, h_1) = (0,0)$, $(t_2, h_2) = (a_1, 0)$, $(t_3, h_3) = (a_1, c + \epsilon)$, and $(t_4, h_4) = (b_1, c + \epsilon)$ for some $c > \epsilon > 0$. Note that the interval $\{\hat{f}_\theta(y \mid x) > c : y \in [0, b_1]\}$ is exactly $[a_1, b_1]$, and that $\int_0^{b_1} \hat{f}_\theta(y \mid x)dy = (c + \epsilon)(b_1 - a_1)$. This exact same procedure is then repeated for each interval, defining $\{t_i, h_i\}$ for $i = 1, \ldots, 4m$. Finally, we add $(t_{4m}, h_{4m}) = (b_m, 0)$ and $(t_{4m+1}, h_{4m+1}) = (b_m, 1)$. If we have more knots (i.e. $m < M$), we set them all to $(1, 0)$ so they have no effect. We have now recreated $\mathcal{P}$. What remains is setting ϵ so that $\int \hat{f}_\theta(y \mid x)dy = 1$. Setting $\epsilon = \frac{1}{|\mathcal{P}|} - c$, the integral is:

$$\int_0^1 \hat{f}_\theta(y \mid x)dy = \sum_{i=1}^m (b_i - a_i)(c + \epsilon) = (c + \epsilon)|\mathcal{P}| = 1, \tag{12}$$

completing the SPICE-ND case.

The SPICE-HPD case. SPICE-HPD requires new knot parameters. Set $(t_1, h_1) = (0,0)$, $(t_2, h_2) = (0, a_i)$, $(t_3, h_3) = (a_1, \epsilon)$, and $(t_4, h_4) = (b_1, \epsilon)$ for some $0 < \epsilon$. Continuing this way, we set $(t_{4(i-1)+1}, h_{4(i-1)+1}) = (t_{4i}, 0)$, $(t_{4(i-1)+2}, h_{4(i-1)+2}) = (a_i, 0)$, $(t_{4(i-1)+3}, h_{4(i-1)+3}) = (a_i, \epsilon)$, and $(t_{4(i-1)+4}, h_{4(i-1)+4}) = (b_i, 0)$ for all $i = 1, \ldots, m$. If $m < M$, we set the remaining knots to $(1, 0)$ so they have no effect. Recall that the prediction set of SPICE-HPD is:

$$\mathcal{C}_{\mathrm{HPD}}(x) = \left\{ y \colon \int_{y' \colon \hat{f}_\theta(y'|x) \leq \hat{f}_\theta(y|x)} \hat{f}_\theta(y' \mid x)dy' > c \right\}, \tag{13}$$

where $c = -\hat{q}$. Since $\hat{f}_\theta = 0$ for any region outside of $\mathscr{P}$, we now no that $\mathcal{C}_{\mathrm{HPD}}(x)$ excludes all the regions out side of $\mathscr{P}$ as desired. Note that $\int \hat{f}_\theta(y \mid x) = \epsilon|(P)|$. Setting $\epsilon = 1/|\mathscr{P}|$ we satisfy the integration to one constraint. It is now also true that for any $y \in \mathscr{P}$, the integral $\int_{y' \in \mathscr{P}} \hat{f}_\theta(y' \mid x)dy' = 1$, since $\hat{f}_\theta(y \mid x) = \epsilon$ for all $y \in \mathscr{P}$. That means $\mathscr{P} = \mathcal{C}_{\mathrm{HPD}}(x)$ as desired.

$\square$

B NEURAL SPLINES

B.1 Knot Parameterization

Throughout this section we will denote the encoding of x as $\mathrm{NN}_{\theta_e}(x) = z$.

B.1.1 Knot Positions

The knot position module maps z to a sorted vector $\{t_i\}_{i=1}^{K}$ where K is the number of knots. We used a modified version of the code of Durkan et al. (2019) to ensure the knot positions were sorted and between zero and one. For numerical reasons, that involved using a minimum distance between knots, ϵ. The first step constructs unnormalized widths from z using a single linear layer. The unnormalized widths are then softmaxed to get widths that sum to one to get widths v_i. Next, non-zero widths w_i are constructed as:

$$w_i = \epsilon + (1 - \epsilon * K)v_i. \tag{14}$$

Widths are converted to positions by taking the cumulative sum, and then prepending zero. Finally, ϵ is added to the last knot position so that it equals one.

SPICE $(n = 2)$ requires one additional position parameter for each pair knot positions. That position parameter is created by taking the midpoint between each consecutive pair of knot positions generated by the above procedure.

B.1.2 SPICE $(n = 1)$ Knot Heights

Unnormalized knot heights are produced using a single linear layer applied to z. The heights are then made positive using the softplus function. Finally, the heights are normalized by dividing the spline defined by the knot positions and heights by their integral (Appendix B.2).

B.1.3 SPICE $(n = 2)$ Knot Heights

The knot heights for each non-midpoint position are the same as for $(n = 1)$, the softplus of a linear layer applied to z. The knot heights for the midpoint positions are generated without using an activation, so they can be negative. Lagrange coefficients (a_i, b_i, c_i) which interpolate between each pair of consecutive positions and heights and the midpoint positions and heights, are computed. These define the polynomial $a_i y^2 + b_i y + c$ on $[t_i, t_{i+1}]$. After zero-truncation, these polynomials define a spline, which is then divided by its integral so it integrates to one.

B.2 Spline Integration

B.2.1 Integrating SPICE $(n = 1)$

Integrating a piecewise linear function just requires summing together the area of a series of trapezoids. Each trapezoid's area is $\frac{1}{2}(t_{i+1} - t_i)(h_i + h_{i+1})$.

B.2.2 Integrating SPICE $(n = 2)$

Between each consecutive knot positions, we have a zero truncated polynomial $\bar{p}_i(y) = \max[a_i y^2 + b_i y + c, 0]$ on $[t_i, t_{i+1}]$. The first step is to integrate the non-zero-truncated polynomial $\left.(a_i/3y^3 + b_i/2y^2 + c_i y)\right|_{t_i}^{t_{i+1}}$. The next step is to compute the roots $r_i^1 < r_i^2$ (if they exist) using the quadratic formula. If they exist, they are clipped to be in $[t_i, t_{i+1}]$. The same integral is then computed between the roots (which is necessarily negative) and then subtracted from the non-zero-truncated integral.

B.3 Building Negative Density Prediction Sets

Given a conformal quantile $\hat{q}$, the first step is to compute all the intersection positions of each polynomial component $\hat{f}_\theta(y \mid x)$ and $-\hat{q}$. The intersections are found by computing the roots of the polynomial $p_i(y) + \hat{q}$. The roots are then clipped to be in $[t_i, t_{i+1}]$.

In the $(n = 1)$ case, whether the interval is inside or outside the clipped roots is determined by the slope of polynomial. In the $(n = 2)$ case, there are two possible cases to consider.

If the roots do not exist (are complex), then the interval is set to $[t_i, t_{i+1}]$ if the vertex of the polynomial is above zero $(c_i - b_i^2/(4a) > 0)$. Otherwise the interval is set to be empty.

If the roots do exist, the sign of the vertex is used to determine whether the prediction interval is between (positive vertex) or outside of the roots (negative vertex). In the outside roots case, we get two intervals, which are $[t_i, r_i^1]$ and $[r_i^2, t_{i+1}]$.

The intervals for each polynomial component are unioned to compute the prediction set $\mathcal{C}_{\mathrm{ND}}(x)$.

B.4   Building HPD Prediction Sets

Given a conformal quantile $\hat{q}$, the prediction set is computed using the bisection algorithm. A lower bound $\ell_0 = 0$ and $u_0 = \sup_y \hat{f}_\theta(y \mid x)$ are initialized. Next a midpoint $m_i = (u_i + \ell_i)/2$ is computed. Next, the following integral is computed:

$$F_i = \int_{\{y:\ \hat{f}_\theta(y|x) < m_i\}} \hat{f}_\theta(y \mid x) dy. \tag{15}$$

If $F_i < -\hat{q}$, then $\ell_{i+1} = m_i$ and $u_{i+1} = u_i$; otherwise $\ell_{i+1} = \ell_i$ and $u_{i+1} = m_i$. This process is repeated 15 times so that $F_15 \approx -\hat{q}$. The HPD-prediction set is then $\{y : \hat{f}_\theta(y \mid x) > F_{15}\}$, which intervals can be computed the same way as for SPICE-ND.

C   EXPERIMENTAL DETAILS

C.1   Data Preprocessing

For each model, the training inputs (x) were preprocessed to have approximately zero mean and standard deviation one using the population mean and standard deviation estimated from the train split.

For SPICE models, the targets, y, were processed to range approximately between zero and one using the scikit-Learn (Pedregosa et al., 2011) min-max scaler fit to the train split ys.

For the baseline models, besides Hist., the ys were transformed to have approximately zero mean and standard deviation one.

For Hist., ys were binned into discrete bins, with the number of bins being a hyperparameter. The bin boundaries were evenly spaced between the minimum and maximum value of y.

C.2   Shared Neural Network Encoder

Every model had a two-layer fully-connected neural network with GeLU (Hendrycks and Gimpel, 2016) activations to encode x. The first layer mapped the input dimension to 32 hidden units. The second layer mapped 32 hidden units to 32 encoding units.

Each model used depth one linear layers following the encoding to predict quantiles / bin probabilities / distribution parameters.

C.3   Benchmark Datasets

We give a brief description of each benchmark dataset and its source. See Romano et al. (2019) for more details about these data.

- BIKE: UCI repository bike sharing data (Fanaee-T, 2013) (CC BY 4.0 license).

- BIO: UCI repository (Rana, 2013) protein property prediction dataset (CC BY 4.0 license).

- BLOG: UCI repository (Buza, 2014) blog feedback prediction dataset (CC BY 4.0 license).

- MEPS19-21: Medical expenditure panel survey datasets numbers 19 through 21 (Cohen et al., 2009) (Open Database License).

- STAR: Student achievement dataset (Achilles et al., 2008) (CC0 1.0 License).

- TEMP.: Cho et al. (2020) introduced temperature forecasting data to compare machine learning models for correcting numerical weather simulations (CC BY 4.0 license).

C.4   Data Splits

The data were randomly split into five buckets: train 50%, validation 10%, calibration 10%, calibration validation 10%, test 20%. The sample assignments within train / validation / calibration / validation calibration were determined by random seed, while the test set was held constant across all experiments and was only used to generate results after hyperoptimization.

C.5   Hyperparameter Optimization

Each model was hyperoptimized over a grid of five learning rates and four values of a capacity hyperparameter for a total of 20 combinations. Each hyperparameter combination was evaluated on three different random seeds for robustness.

Evaluation was done by conformal calibration on the calibration split and conformal prediction on the calibration validation split. Hyperparameters were selected uniquely for each dataset and model to minimize the average size at 90% nominal coverage averaged across all three seeds.

The hyperparameter grids were as follows:

1. **CQR**. Learning rate: $[1e-1, 5e-2, 1e-2, 5e-3, 1e-3]$. Nominal quantile interval: $[0.3, 0.5, 0.7, 0.9]$.

2. **CHR**. Learning rate: $[1e-1, 5e-2, 1e-2, 5e-3, 1e-3]$. Number of quantiles: $[50, 200, 350, 500]$.

3. **PCP**. Learning rate: $[1e-1, 5e-2, 1e-2, 5e-3, 1e-3]$. Number of mixture components: $[5, 10, 15, 20]$.

4. **Hist**. Learning rate: $[1e-1, 5e-2, 1e-2, 5e-3, 1e-3]$. Number of bins: $[11, 21, 31, 51]$.

5. **SPICE** ($n = 1$). Learning rate: $[5e-2, 1e-2, 5e-3, 1e-3, 5e-4]$. Number of knots: $[11, 21, 31, 51]$.

6. **SPICE** ($n = 2$). Learning rate: $[1e-2, 5e-3, 1e-3, 5e-4, 1e-4]$. Number of knots: $[11, 21, 31, 51]$.

C.6   Model Training

Models were trained using the AdamW (Loshchilov and Hutter, 2019) optimizer with weight decay $1e-4$ for a maximum of 50,000 batches of size 512 stopping earlier if the validation loss did not improve after 125 passes over the training data. The final model was selected based on its best validation loss, which was calculated every 100 batches or full pass over the training data, whichever was smaller. The validation loss was calculated on 10 batches or the full validation set, whichever was smaller. The learning rate was decayed according to the cosine annealing schedule (Loshchilov and Hutter, 2017). The gradient was clipped to have max-norm of five.

C.7   Conditional Histogram Baseline

The Hist. model was trained as a classifier model on discretized y values (Appendix C.1) using the negative log-likelihood loss. The architecture and hyperparameter optimization was shared with the other models. The conformal score was the negative probability of the true class.

C.8   Prediction Set Size Normalization

The size normalization can be thought of as dividing by the prediction set size of an unconditional version of the Hist. baseline. y was first discretized into 20 evenly spaced bins, then the probability of y being in each bin was computed on the train split. This resulted in an unconditional "model" for predicting $y \mid x$. This model was then conformalized using the calibration split with the negative-probability of y as the conformal score. The score was then used to compute a constant prediction set for the test data. The size of that prediction set was the normalization constant for that dataset. All prediction set sizes in the results were divided by the corresponding normalization constant for the relevant dataset. This process is in the function called `get_baseline_size` in the supplementary code.

C.9   Label Conditional Coverage Approximation

Label conditional coverage was approximated by binning Y into five buckets. Coverage was computed on the test set for each bucket. Label conditional size was reported as the worst coverage across all five buckets.

C.10   Synthetic Data Experiment

In order to generate the synthetic bimodal data, 2,000 x values were taken at even spacing between zero and one. Noise was then sampled as

$$\sigma(x) = 0.1 + xU,$$

where U was random uniform on $[0, 1]$ for each x. A random indicator was then sampled for each x with equal probability as zero or one. If the indicator was one, y was set to σ. Otherwise y was set to $-\sigma$. Preprocessing was done for each model the same way as for the benchmark data (Appendix C.1).

D   ADDITIONAL RESULTS

D.1   Worst-slab Coverage (WSC)

WSC is an approximation of conditional coverage that involves an adversarial optimization problem to find the minimum coverage over a slab in feature space. We used the implementation of Romano et al. (2020), but found some unexpected results, such as WCS being greater than marginal coverage (PCP, STAR dataset). This indicates a problem in the optimization procedure, which led us to prefer the simpler label-conditional coverage approximation. The model ranks remain mostly the same for conditional coverage according to WSC, with SPICE-HPD ($n = 1$) having the best conditional coverage and SPICE-HPD ($n = 2$) and PCP being second best.

Table 4: Worst-slab approximated conditional coverage results. The nominal coverage was set to 90%. **Bold** indicates the best mean; <u>underline</u> indicates the second best. MEAN is the average size across all datasets.

MODEL	BIKE	BIO	BLOG	MEPS19	MEPS20	MEPS21	STAR	TEMP.	MEAN
CQR	0.66 ± 0.01	0.67 ± 0.00	0.68 ± 0.00	0.67 ± 0.00	0.65 ± 0.00	0.54 ± 0.01	0.89 ± 0.01	0.61 ± 0.01	0.67 ± 0.03
CHR	0.66 ± 0.01	0.69 ± 0.00	0.69 ± 0.00	0.67 ± 0.00	0.66 ± 0.00	0.55 ± 0.00	0.88 ± 0.01	0.58 ± 0.01	0.67 ± 0.03
PCP	0.81 ± 0.01	0.87 ± 0.00	$\underline{0.76 \pm 0.00}$	$\underline{0.80 \pm 0.01}$	$\underline{0.82 \pm 0.01}$	$\underline{0.78 \pm 0.01}$	$\mathbf{0.94 \pm 0.01}$	$\underline{0.90 \pm 0.01}$	$\underline{0.83 \pm 0.02}$
Hist.	0.83 ± 0.01	$\underline{0.89 \pm 0.00}$	0.65 ± 0.01	0.75 ± 0.01	0.50 ± 0.06	0.59 ± 0.01	$\underline{0.92 \pm 0.01}$	0.89 ± 0.01	0.75 ± 0.06
SPICE-ND ($n = 1$)	0.83 ± 0.01	0.88 ± 0.00	0.69 ± 0.00	0.74 ± 0.01	0.77 ± 0.00	0.67 ± 0.01	0.89 ± 0.02	0.89 ± 0.01	0.79 ± 0.03
SPICE-HPD ($n = 1$)	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.89 \pm 0.00}$	0.76 ± 0.01	$\mathbf{0.88 \pm 0.01}$	$\mathbf{0.90 \pm 0.01}$	$\mathbf{0.89 \pm 0.01}$	0.88 ± 0.02	0.88 ± 0.01	$\mathbf{0.87 \pm 0.02}$
SPICE-ND ($n = 2$)	0.82 ± 0.01	0.88 ± 0.01	0.71 ± 0.00	0.73 ± 0.01	0.74 ± 0.01	0.63 ± 0.02	0.88 ± 0.01	$\mathbf{0.90 \pm 0.01}$	0.79 ± 0.03
SPICE-HPD ($n = 2$)	$\underline{0.84 \pm 0.01}$	0.88 ± 0.00	$\mathbf{0.85 \pm 0.00}$	0.78 ± 0.01	0.76 ± 0.02	0.73 ± 0.02	0.90 ± 0.01	0.88 ± 0.01	0.83 ± 0.02