Supplementary Material: Improving Conditional Coverage via Orthogonal Quantile Regression

Shai Feldman
Department of Computer Science
Technion, Israel
`shai.feldman@cs.technion.ac.il`

Stephen Bates
Departments of Statistics and of EECS
UC Berkeley
`stephenbates@cs.berkeley.edu`

Yaniv Romano
Departments of Electrical and Computer Engineering
and of Computer Science
Technion, Israel
`yromano@technion.ac.il`

S1 Emperical estimations

S1.1 A differentiable coverage identifier

Given a prediction interval $\hat{C}(X_i) = [\hat{q}_{\alpha_{\text{lo}}}(X_i), \hat{q}_{\alpha_{\text{lo}}}(X_i)]$ for a point X_i, we approximate its coverage indicator $\mathbb{1}[Y \in \hat{C}(X_i)]$ in the following way:

$$\begin{aligned}\tilde{V}_i &= \tanh\left(c \min\{Y_i - \hat{q}_{\alpha_{\text{lo}}}(X_i), \hat{q}_{\alpha_{\text{hi}}}(X_i) - Y_i\}\right)\\ \hat{V}_i &= \frac{1}{2}\left(\tilde{V}_i + 1\right)\end{aligned} \tag{S1}$$

where $c \in \mathbb{R}^+$ controls the slope of the step function. In the experiments, we set c to be equal to $5 \cdot 10^3$. This approximation is differentiable and used in practice.

S2 Theoretical results

S2.1 Proof of proposition 1

Proof. Consider an $l \in \mathbb{R}$, and $v \in \{0, 1\}$. To prove the theorem, it suffices to show that $\mathbb{P}(V = v \mid L = l) = \mathbb{P}(V = v)$. Denote $g_X, g_{X|L}$ the density functions of X and $X|L$ respectively, and denote $\beta = \mathbb{P}(V = v \mid X = x)$. It is given that the interval constructed satisfies $\mathbb{P}(V = 1 \mid X = x) = 1 - \alpha$ for all $x \in \mathcal{X}$, and therefore β is a constant and equals $1 - \alpha$ if $v = 1$, or α if $v = 0$.

First, we show that $\mathbb{P}(V = v) = \beta$:

$$\mathbb{P}(V = v) = \int_{x \in \mathcal{X}} \mathbb{P}(V = v \mid X = x) g_X(x)\, dx = \beta \int_{x \in \mathcal{X}} g_X(x)\, dx = \beta$$

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

To conclude our proof, we show that $\mathbb{P}(V = v \mid L = l) = \beta$ as well.

$$\begin{aligned}
\mathbb{P}(V = v \mid L = l) &= \int_{\{x \in \mathcal{X} \mid L = l\}} \mathbb{P}(V = v \mid L = l, X = x) g_{X|L}(x \mid l)\, dx \\
&= \int_{\{x \in \mathcal{X} \mid L = l\}} \mathbb{P}(V = v \mid X = x) g_{X|L}(x \mid l)\, dx \\
&= \int_{\{x \in \mathcal{X} \mid L = l\}} \beta g_{X|L}(x \mid l)\, dx \\
&= \beta = \mathbb{P}(V = v)
\end{aligned}$$

□

S2.2 Proof of corollary

Proof. The interval constructed by true conditional quantiles $C(X) = [q_{\alpha_{\text{lo}}}(X), q_{\alpha_{\text{hi}}}(X)]$ satisfies $\mathbb{P}(Y \in C(X) | X = x) = 1 - \alpha$ for all $x \in \mathcal{X}$, and hence, from Proposition 1 we conclude that $L \perp\!\!\!\perp V$.

□

S2.3 Proof of theorem 1

Proof. The true quantiles minimize the first term in the sum, by the properties of pinball loss or interval score loss. Moving to the next term. Assuming $Y \mid X$ is continuous, from Corollary 1 the true conditional quantiles satisfy: $V \perp\!\!\!\perp L$, and this implies that their correlation, denoted as $\text{CORR}(L, V)$, and HSIC are fixed and equal zero, and therefore:

$$\begin{aligned}
&\text{CORR}(L, V) = 0 \Rightarrow \mathcal{R}_{\text{corr}}(L, V) = |\text{CORR}(L, V)| = |0| = 0, \\
&\text{HSIC}(L, V) = 0 \Rightarrow \mathcal{R}_{\text{HSIC}}(L, V) = \sqrt{\text{HSIC}(L, V)} = \sqrt{0} = 0.
\end{aligned}$$

This means that for either choice loss function, $\mathcal{R}(L, V) = 0$ which is the minimal value of the second term, as it is non-negative. Therefore, the true quantiles are a solution to the minimization problem, since they minimize both terms separately.

Turning to the uniqueness, assume there is a unique solution for $\gamma = 0$. Then, since the true quantiles minimize the the pinball loss or interval score loss, it follows that the true conditional quantiles are the unique solution. We proved that they attain $\mathcal{R}(L, V) = 0$, the minimum value this term could achieve. Thus, we conclude that there is only one solution for all $\gamma > 0$. □

S3 Datasets details

S3.1 Synthetic datasets details

The synthetic dataset is determined by the parameter λ which controls the variance of the response value of the minority group. The generation of the feature vector and the response variable is done in the following way:

$$
\begin{aligned}
&\hat{\beta} \sim \text{Uniform}(0,1)^{50}, \hat{\gamma} \sim \text{Uniform}(0,1)^{50}, \\
&\beta = \frac{\hat{\beta}}{\|\hat{\beta}\|_2}, \gamma = \frac{\hat{\gamma}}{\|\hat{\gamma}\|_2}, \\
&\varepsilon_{1,i} \sim \mathcal{N}(0,1), \varepsilon_{2,i} \sim \mathcal{N}(0,1), && 1 \le i \le n, \\
&X_{i,1-49} \sim \text{Uniform}(0,5)^{49}, && 1 \le i \le n, \\
&X_{i,0} = \begin{cases} 0, & \text{w.p. } 0.8 \\ 1, & \text{otherwise} \end{cases} && 1 \le i \le n, \\
&Y_i = \begin{cases} 0.03\beta^T X_i \varepsilon_{1,i}, & X_{i,0} = 0 \\ 0.03\gamma^T X_i \varepsilon_{1,i} + \lambda \varepsilon_{2,i}, & X_{i,0} = 1 \end{cases} && 1 \le i \le n,
\end{aligned}
$$

where Uniform(a, b) is a uniform distribution on the interval (a, b), and $\mathcal{N}(0, 1)$ is the standard Gaussian distribution. The dataset with minority noise level set to low was created with $\lambda = 3$, and the one with the High noise level with $\lambda = 10$. Both datasets contain 7000 samples, and were generated with a seed value equals to 1.

S3.2 Real dataset details

Table S1 shows the size of each data set and the feature dimension.

Table S1: Real datasets information. Number of samples and feature dimension of each real dataset we used in the experiments

Dataset Name	Number of Samples	Feature Dimension
facebook_1 [1]	40948	53
facebook_2 [1]	81311	53
blog_data [2]	52397	2805
bio [3]	45730	9
kin8nm [4]	8192	8
naval [5]	11934	17
meps_19 [6]	15785	139
meps_20 [7]	17541	139
meps_21 [8]	15656	139

S4 Experimental setup

S4.1 Setup

The network we used receives as an input a vector of size $p + 1$ (where the feature dimention is p). The first p variables in the input vector correspond to the elements of the feature vector, and the last variable is the quantile level of the desired quantile. We trained quantile regression with pinball loss over two quantile levels: 95% and 5% level quantiles, and trained quantile regression with interval score over all quantile levels. For each model, we built prediction intervals using the 95%-th and 5%-th quantiles it outputted, as explained in 2.1. The code we used is based on the implementation of [9]. The penalty multipliers for each dataset were chosen by an independent train-validation-test split (with seed=42), so the coefficient that achieved the best performance was taken. The multipliers tested for the real data are: 0.1, 0.5, for pinball loss, and 0.1, 0.5, 1, 3, for interval score loss. For the synthetic data we checked: 0.1, 0.5 for both losses. When combining our penalty with pinball loss, the coefficient given to the model is multiplied by 0.1.

S4.2 Synthetic data experiments

We split the synthetic datasets into a training set (64%), a validation set (16%) for early stopping, and a test set (20%) to evaluate performance. After that, the feature vectors and the labels were preprocessed using z-score normalization. The neural network is made of 2 layers of 64 hidden units, and a ReLU activation function. The network does not contain a dropout layer. The learning rate used is $1e^{-3}$, the model's optimizer is Adam [10], and the batch size is 1024 for all methods. The maximum number of epochs is 10000, but the training is stopped early if the validation loss does not increase for 200 epochs, and in this case the model with the lowest loss is taken as the final model. The results were averaged over all seeds in the range between 0 and 29 (inclusive).

The decorrelation coefficients used in the experiments are: for pinball loss, 0.5 for both $\lambda = 3$, and $\lambda = 10$, and for interval score loss 3 for both $\lambda = 3, 10$.

S4.3 Real data experiments

First, the datasets: facebook_1, facebook_2, blog_data, and bio, were log scaled: $y = \log(y - \min(y) + 1)$. We split each dataset it into a training set (54%), a validation set (6%) for early stopping, and a test set (40%) to evaluate performance. After that, the feature vectors and the labels were preprocessed using z-score normalization. The neural network is made of 3 layers of 64 hidden units, and ReLU activation function. The network contains a dropout layer with parameter 0.1. The learning rate, training strategy and batch size are the same as described in S4.2. The maximum number of epochs is 10000, and we used the same early stopping technique described in S4.2. The results were averaged over all seeds in the range between 0 and 29 (inclusive).

Figure 2 was produced by splitting all test's coverages and lengths over all seeds to hundred bins, and averaging each bin separately. Both `vanilla QR` and `orthogonal QR` used pinball loss as an objective function, and the penalty used by our method is $\mathcal{R}_{\text{corr}}$.

Table S2 displays the multiplier used for each data set and method.

Table S2: Penalty multipliers used for each dataset

Pinball Loss

Dataset Name	Decorr multiplier	HSIC multiplier
facebook_1	0.5	0.5
facebook_2	0.5	0.5
blog_data	0.5	0.5
bio	0.1	0.1
kin8nm	0.1	0.1
naval	0.1	0.1
meps_19	0.5	0.1
meps_20	0.5	0.1
meps_21	0.5	0.5

Interval Score Loss

Dataset Name	Decorr multiplier
facebook_1	0.5
facebook_2	0.5
blog_data	1
bio	0.1
kin8nm	0.5
naval	0.1
meps_19	3
meps_20	3
meps_21	3

S4.4 Conformalized quantile regression experiments

We used the same setting as in S4.3, except for the following changes. The dataset was split into a training set (54%), a validation set (6%) for early stopping, a calibration set (20%) to achieve valid marginal coverage, and a test set (20%) to evaluate performance. The calibration method used is Conformalized Quantile Regression [11].

S4.5 Machine's spec

The resources used for the experiments are:

- **CPU**: Intel(R) Core(TM) i5-10600K CPU 4.10GHz.
- **GPU**: NVIDIA GeForce RTX 2060 SUPER.

- **OS**: Windows 10.

S5 Additional results

S5.1 Synthetic data

S5.1.1 Interval score loss results

As shown in Table S3, when combined with interval score loss, the suggested penalty consistently improves all metrics, and balances the coverage rates over the majority and minority subgroups.

Table S3: Simulated data experiments - using interval score loss with either `QR` (baseline) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{corr}}$. Refer to the caption of Table 1 for further details. The standard errors for coverage and width are about 0.45, 0.1, respectively. See Table S10 for a full reporting of all standard errors.

Minority Group Uncertainty	**Majority Coverage (%)**	**Minority Coverage (%)**	**Majority Lengths**	**Minority Lengths**	**Improvement (%)**		
	baseline / `OQR`	baseline / `OQR`	baseline / `OQR`	baseline / `OQR`	`corr`	`HSIC`	Δ`WSC`
Low	84.38 / 88.56	72.77 / 78.41	1.73 / 1.89	7.46 / 8.70	+35.03	+46.48	+49.80
High	85.99 / 89.89	74.08 / 80.05	2.22 / 2.53	24.99 / 29.07	+27.91	+7.00	+17.62

S5.1.2 Weighted quantile regression

Tables S4, S5 show that combining the proposed decorrelation penalty with `weighted QR` improves the conditional coverage, according to the examined metrics. Specifically, while the values of `corr` attained by `weighted QR` and its orthogonal variant (`OWQR`) are similar, the latter achieves better performance in terms of the HSIC measure (that quantifies non linear relationships between L and V). Observe that our method also improves the group coverage, as the coverage rate of the minority group is closer to the nominal level compared to the baseline approach. The weights we assign to the samples are the absolute residuals of a regression model fitted to the training set (minimizing the mean squared error). We have examined various options for choosing the weight function, including absolute/squared residuals, and the inverse of the absolute/squared residuals. Another option we tested is to assign weights that balance the majority and minority groups (all minority samples are weighted with a factor of 4, and all majority samples are weighted by 1). Among all options, the absolute residual weighting function achieves the best results. In short, this experiment reveals that adding the orthogonality loss to this best version of `weighted QR` improves conditional coverage as well.

S5.1.3 The orthogonality loss coefficient

In Figure S1 we demonstrate the effect of the `Pearson's corr` loss coefficient γ on the resulted intervals. This figure shows that the proposed loss increases the variability of the resulted intervals: it widens the intervals of the minority group, and shortens those of the majority group, improving conditional coverage. As a result, the variability increases but in a non-trivial way: the increased adaptivity we gain improves conditional coverage.

Table S4: Simulated data experiments. Performance of neural network quantile regression, using either `WQR` (baseline) or `orthogonal weighted QR` (`OWQR`) with penalty term $\mathcal{R}_{\text{corr}}$ with coefficient $\gamma = 3$. The performance are evaluated over 60 independent trials. The standard errors for coverage and length are about 2.1, 0.16, respectively. The conditional coverage metrics and their standard errors are presented in Table S5.

Minority Noise Level	**Majority Coverage (%)**	**Minority Coverage (%)**	**Majority Lengths**	**Minority Lengths**
	baseline / `OWQR`	baseline / `OWQR`	baseline / `OWQR`	baseline / `OWQR`
Low	90.87/ 91.87	69.04 / 84.25	6.02 / 5.02	7.33 / 10.75
High	90.20 / 94.73	68.89 / 87.58	18.35 / 15.76	23.56 / 36.18

Table S5: Simulated data: Average metric value (standard error) - using pinball loss with either weighted quantile regression (`WQR`) or `orthogonal WQR` (`OWQR`) with penalty term $\mathcal{R}_{\text{corr}}$. The conditional coverage metrics are described in Section 4. See more details in Table S4.

Minority Group Uncertainty	`corr`		`HSIC`		Δ`WSC`	
	`WQR`	`OWQR`	`WQR`	`OWQR`	`WQR`	`OWQR`
Low	0.049 (0.004)	0.054 (0.005)	7e-4 (4e-5)	4e-4 (4e-5)	1.744 (0.202)	1.839 (0.198)
High	0.066 (0.007)	0.073 (0.006)	0.001 (6e-5)	2e-4 (1e-5)	2.725 (0.215)	1.885 (0.163)

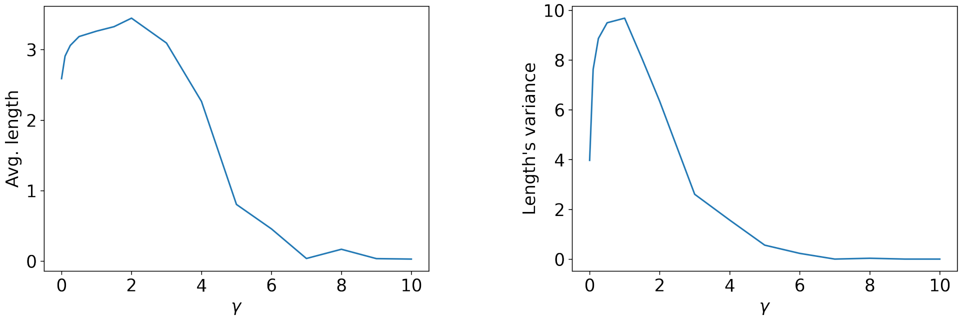

Figure S1: The effect of the `Pearson's corr` loss coefficient γ on the interval's length and its variance on the synthetic data with $\lambda = 3$.

S5.2 Real data

The advantages of adding the suggested orthogonal loss to the interval score loss are summarized in Table S6. Similarly to pinball loss, our regularizer improves the baseline method over all metrics in most datasets. Also, the improvement over Δ`WSC` and Δ`Node-Coverage` metrics suggest that our loss does help to better approximate conditional validity with interval score as well as pinball loss. Overall, we can conclude that our proposal is useful for various scoring rules.

Table S6: Real data experiments - using interval score loss with either `QR` (baseline) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{corr}}$. Refer to the caption of Table 2 for further details. The standard errors for coverage and width are about 0.6, 0.1, respectively. See Table S12 for a full reporting of all standard errors.

Dataset Name	**Coverage (%)**		**Length**		**Improvement (%)**				
	baseline	`OQR`	baseline	`OQR`	`corr`	`HSIC`	Δ`WSC`	Δ`ILS`	Δ`Node`
facebook_1	89.03	91.97	1.43	1.45	+79.02	+96.90	+15.91	+68.69	+1.15
facebook_2	88.82	93.35	1.35	1.38	+90.96	+98.71	+30.53	+69.23	+5.81
blog_data	82.94	87.25	1.58	1.65	+86.13	+94.47	+2.79	+89.66	+18.64
bio	89.93	89.76	2.19	2.19	+22.89	+37.16	-8.85	+48.56	-15.18
kin8nm	90.51	91.81	1.49	1.59	+28.63	+49.19	+13.67	+73.32	+22.77
naval	91.95	92.36	1.95	1.98	+14.18	+30.95	+20.33	+59.16	+26.28
meps_19	83.68	86.16	0.95	1.13	+17.15	+17.49	+24.28	+82.29	+27.97
meps_20	84.86	86.81	1.04	1.21	-10.31	-9.51	+1.36	+77.66	+24.69
meps_21	85.27	86.63	0.98	1.14	+20.88	-49.49	+17.14	+77.89	+10.89

HSIC results

Table S7 presents the results of `vanilla QR` and `orthogonal QR` using HSIC penalty. Similarly to the decorrelation penalty, the HSIC approach also improves the conditional coverage metrics in most data sets.

Table S7: Real data experiments - using pinball loss with either `vanilla QR` (baseline) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{HSIC}}$. Refer to the caption of Table 2 for further details. The standard errors for coverage and width are about 0.5, 0.06, respectively. See Table S13 for a full reporting of all standard errors.

Dataset Name	**Coverage (%)**		**Length**		**Improvement (%)**				
	baseline	`OQR`	baseline	`OQR`	`corr`	`HSIC`	Δ`WSC`	Δ`ILS`	Δ`Node`
facebook_1	88.10	93.86	1.09	1.43	+73.51	+88.41	+32.76	+76.86	+38.15
facebook_2	87.38	94.77	1.07	1.37	+88.49	+98.77	+42.68	+82.77	+57.24
blog_data	82.89	92.88	1.36	1.64	+63.79	+35.79	+32.58	+89.66	+41.74
bio	88.42	89.47	1.88	2.03	+11.26	+16.18	-42.33	+43.76	-25.75
kin8nm	84.63	88.22	0.98	1.20	+9.00	+25.34	-42.92	+52.75	+27.39
naval	89.89	89.72	0.56	1.21	+49.73	+5.52	-51.05	+61.98	+8.52
meps_19	82.44	85.47	0.84	0.94	+9.21	-8.17	+12.26	+78.54	+19.16
meps_20	82.81	86.02	0.86	0.96	-20.63	-34.17	+2.57	+81.03	+14.74
meps_21	82.50	82.51	0.86	0.94	-19.50	+10.39	+8.00	+84.43	+24.41

Table S8 compares the two proposed loss functions, and displays the improvement obtained by regularizing with decorrelation instead of HSIC. As expected, the method optimizing `Pearson's corr` consistently achieves better results over this metric compared to pinball loss combined with HSIC regularizer. Surprisingly, in most datasets the model that uses decorrelation penalty attains a better `HSIC` value than the one optimizing directly this metric. These empirical drawbacks of HSIC regularization compared to our suggested penalty might have been caused by the fact that the latter receives twice as many samples in each gradient step, or its coefficients were more tuned. Nevertheless, the method using HSIC is able to achieve better statistical efficiency, which is probably due to its strength. Overall, we can conclude that thanks to the low computational burden, it is easier to fine tune the coefficients of the decorrelation loss and use more samples, but it might not be as strong as HSIC.

Table S8: Real data experiments. Performance of a neural network model for quantile regression, `orthogonal QR` using either ($\mathcal{R}_{\text{HSIC}}$) penalty or ($\mathcal{R}_{\text{corr}}$) penalty. Refer to the caption of Table 2 for further details. The standard errors for coverage and width are about 0.5, 0.06, respectively. See Tables S11,S13 for a full reporting of all standard errors.

Dataset Name	**Coverage (%)**		**Length**		**Improvement (%)**		
	$\mathcal{R}_{\text{HSIC}}$	$\mathcal{R}_{\text{corr}}$	$\mathcal{R}_{\text{HSIC}}$	$\mathcal{R}_{\text{corr}}$	`corr`	`HSIC`	Δ`WSC`
facebook_1	93.86	90.48	1.43	1.44	+28.59	-100.51	+0.98
facebook_2	94.77	91.13	1.37	1.41	+24.18	-195.60	-15.38
blog_data	92.88	88.92	1.64	1.64	+37.84	-52.21	-16.51
bio	89.47	89.08	2.03	2.03	+47.59	+70.83	-1.69
kin8nm	88.22	88.62	1.20	1.28	+20.38	+30.98	+40.50
naval	89.72	89.50	1.21	1.49	+50.64	+30.12	-4.28
meps_19	85.47	85.16	0.94	1.00	+49.51	+28.12	+35.35
meps_20	86.02	84.27	0.96	1.03	+53.17	+26.95	+22.05
meps_21	82.51	84.07	0.94	0.99	+64.40	+3.60	+13.89

S6 Standard error results

S6.1 Synthetic data

In the following tables we present the mean value and standard errors received for each metric in the synthetic data experiments.

Table S9: Simulated data: Average metric value (standard error) - using pinball loss with either `vanilla QR` (`QR`) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{corr}}$.

Minority Group Uncertainty	`corr`		`HSIC`		Δ`WSC`	
	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`
Low	.105 (.005)	**.038 (.008)**	.001 (1e-5)	**1e-4 (1e-5)**	**2.195 (.337)**	2.500 (.373)
High	.115 (.006)	**.032 (.006)**	1e-4 (1e-5)	**1e-4 (1e-4)**	3.240 (.384)	**2.658 (.335)**

Table S10: Simulated data: Average metric value (standard error) - using interval score loss with either quantile regression (`QR`) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{corr}}$.

Minority Group Uncertainty	`corr`		`HSIC`		Δ`WSC`	
	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`
Low	.094 (.008)	**.061 (.008)**	1e-4 (1e-4)	**1e-4 (1e-5)**	3.062 (.322)	**1.537 (.339)**
High	.124 (.009)	**.089 (.014)**	1e-4 (1e-5)	**1e-4 (1e-4)**	3.196 (.369)	**2.633 (.359)**

S6.2 Real data

In the following tables, we present the mean value and standard errors received for each metric in the real data experiments. Table S15 reports the results of quantile regression forests on the real datasets.

Table S11: Real data: Average metric value (standard error) - using pinball loss with either `vanilla QR` (`QR`) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{corr}}$.

Dataset Name	`corr`		`HSIC`		Δ`WSC`		Δ`ILS`		Δ`Node`	
	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`
facebook_1	.057 (.013)	**.011 (.002)**	1e-4 (1e-4)	**1e-5 (1e-6)**	6.371 (.731)	**4.242 (.399)**	9.345 (.520)	**3.195 (.185)**	3.555 (.491)	**1.874 (.238)**
facebook_2	.099 (.028)	**.009 (.002)**	.002 (.001)	**1e-5 (1e-6)**	6.810 (1.239)	**4.504 (.314)**	10.104 (.986)	**3.465 (.117)**	4.654 (1.433)	**2.065 (.284)**
blog_data	.043 (.002)	**.010 (.002)**	1e-5 (1e-5)	**1e-5 (1e-5)**	10.037 (.330)	**7.885 (.306)**	12.773 (.261)	**1.285 (.178)**	4.823 (.386)	**3.163 (.322)**
bio	.084 (.002)	**.039 (.002)**	1e-4 (1e-5)	**1e-5 (1e-6)**	**1.218 (.192)**	1.762 (.248)	8.461 (.263)	**4.806 (.167)**	.912 (.153)	**.846 (.116)**
kin8nm	.283 (.003)	**.205 (.006)**	.002 (1e-5)	**.001 (1e-5)**	1.638 (.216)	**1.393 (.205)**	16.770 (.452)	**7.148 (.283)**	**1.471 (.256)**	1.713 (.347)
naval	.269 (.006)	**.067 (.006)**	.001 (1e-5)	**1e-4 (1e-5)**	**2.824 (.394)**	4.449 (.563)	8.674 (.633)	**2.398 (.278)**	**2.410 (.312)**	2.756 (.427)
meps_19	.056 (.006)	**.026 (.004)**	1e-4 (1e-6)	**1e-5 (1e-5)**	8.580 (.644)	**4.866 (.496)**	12.976 (.551)	**1.550 (.204)**	7.204 (1.066)	**3.931 (.584)**
meps_20	.048 (.005)	**.027 (.005)**	1e-5 (1e-6)	**1e-5 (1e-5)**	7.269 (.697)	**5.520 (.590)**	11.351 (.459)	**1.735 (.270)**	5.328 (.850)	**2.830 (.494)**
meps_21	.064 (.006)	**.027 (.004)**	1e-4 (1e-5)	**1e-4 (1e-5)**	8.745 (.687)	**6.927 (.738)**	12.940 (.507)	**1.938 (.236)**	6.043 (.911)	**3.730 (.569)**

References

[1] Facebook comment volume data set. `https://archive.ics.uci.edu/ml/datasets/Facebook+Comment+Volume+Dataset`. Accessed: January, 2019.

[2] Blogfeedback data set. `https://archive.ics.uci.edu/ml/datasets/BlogFeedback`. Accessed: January, 2019.

Table S12: Real data: Average metric value (standard error) - using interval score loss with either quantile regression (`QR`) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{corr}}$.

Dataset Name	`corr`		`HSIC`		Δ`WSC`		Δ`ILS`		Δ`Node`	
	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`
facebook_1	.075 (.020)	**.016 (.003)**	.001 (.001)	**1e-5 (1e-6)**	3.392 (1.079)	**2.853 (.405)**	7.230 (.661)	**2.264 (.190)**	1.189 (.190)	**1.175 (.168)**
facebook_2	.083 (.028)	**.007 (.002)**	.002 (.002)	**1e-5 (1e-6)**	4.161 (1.483)	**2.891 (.227)**	6.707 (.660)	**2.063 (.131)**	2.065 (.708)	**1.945 (.330)**
blog_data	.081 (.026)	**.011 (.002)**	.001 (1e-4)	**1e-5 (1e-6)**	10.468 (.450)	**10.176 (.656)**	10.610 (.338)	**1.097 (.180)**	4.875 (.657)	**3.966 (.670)**
bio	.089 (.004)	**.069 (.004)**	1e-4 (1e-5)	**1e-4 (1e-5)**	**1.761 (.167)**	1.917 (.204)	7.853 (.684)	**4.040 (.290)**	**.796 (.154)**	.917 (.137)
kin8nm	.250 (.004)	**.178 (.006)**	.001 (1e-5)	**1e-4 (1e-5)**	1.740 (.234)	**1.502 (.231)**	14.186 (.823)	**3.785 (.276)**	1.775 (.286)	**1.371 (.212)**
naval	.157 (.003)	**.135 (.005)**	1e-4 (1e-5)	**1e-4 (1e-5)**	4.189 (.601)	**3.337 (.560)**	9.506 (.937)	**3.883 (.378)**	1.450 (.233)	**1.069 (.166)**
meps_19	.040 (.010)	**.033 (.005)**	1e-4 (1e-5)	**1e-4 (1e-5)**	10.867 (1.001)	**8.228 (1.018)**	11.066 (1.220)	**1.960 (.288)**	6.720 (.917)	**4.841 (.659)**
meps_20	**.028 (.005)**	.031 (.005)	**1e-4 (1e-5)**	1e-4 (1e-5)	7.898 (.430)	**7.790 (.634)**	8.513 (.476)	**1.902 (.222)**	4.234 (.763)	**3.189 (.565)**
meps_21	.047 (.004)	**.037 (.004)**	**1e-4 (1e-5)**	1e-4 (1e-5)	9.295 (.668)	**7.702 (.789)**	9.743 (.459)	**2.154 (.258)**	5.771 (1.112)	**5.143 (.820)**

Table S13: Real data: Average metric value (standard error) - using pinball loss with either `vanilla QR` (`QR`) or `orthogonal QR` (`OQR`) with penalty term $\mathcal{R}_{\text{HSIC}}$.

Dataset Name	`corr`		`HSIC`		Δ`WSC`		Δ`ILS`		Δ`Node`	
	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`	`QR`	`OQR`
facebook_1	.057 (.013)	**.015 (.003)**	1e-4 (1e-4)	**1e-5 (1e-6)**	6.371 (.731)	**4.284 (.357)**	9.620 (.521)	**2.226 (.141)**	3.300 (.552)	**2.041 (.346)**
facebook_2	.099 (.028)	**.011 (.002)**	.002 (.001)	**1e-5 (1e-6)**	6.810 (1.239)	**3.904 (.247)**	11.088 (1.170)	**1.910 (.103)**	5.849 (1.887)	**2.501 (.355)**
blog_data	.043 (.002)	**.016 (.004)**	1e-5 (1e-5)	**1e-5 (1e-6)**	10.037 (.330)	**6.768 (.270)**	13.524 (.286)	**1.399 (.102)**	4.851 (.467)	**2.826 (.403)**
bio	.084 (.002)	**.075 (.002)**	1e-4 (1e-5)	**1e-4 (1e-5)**	**1.218 (.192)**	1.733 (.239)	8.530 (.210)	**4.797 (.107)**	**.772 (.130)**	.971 (.162)
kin8nm	.283 (.003)	**.257 (.004)**	.002 (1e-5)	**.001 (1e-5)**	**1.638 (.216)**	2.341 (.326)	17.769 (.500)	**8.396 (.189)**	1.813 (.327)	**1.317 (.212)**
naval	.269 (.006)	**.135 (.006)**	.001 (1e-5)	**.001 (1e-5)**	**2.824 (.394)**	4.266 (.513)	6.480 (.640)	**2.464 (.273)**	2.374 (.380)	**2.172 (.327)**
meps_19	.056 (.006)	**.051 (.006)**	**1e-4 (1e-6)**	1e-4 (1e-5)	8.580 (.644)	**7.528 (.632)**	11.940 (.299)	**2.563 (.273)**	6.736 (1.050)	**5.446 (.854)**
meps_20	**.048 (.005)**	.057 (.008)	**1e-5 (1e-6)**	1e-4 (1e-5)	7.269 (.697)	**7.082 (.655)**	11.128 (.468)	**2.111 (.274)**	4.979 (.916)	**4.245 (.800)**
meps_21	**.064 (.006)**	.077 (.014)	1e-4 (1e-5)	**1e-4 (1e-5)**	8.745 (.687)	**8.045 (.754)**	12.333 (.507)	**1.920 (.283)**	3.899 (.673)	**2.947 (.528)**

Table S14: Real data: Average metric value (standard error) - using pinball loss with conformalization and either `vanilla CQR` (`CQR`) or `orthogonal CQR` (`COQR`) with penalty term $\mathcal{R}_{\text{corr}}$.

Dataset Name	`corr`		`HSIC`		Δ`WSC`		Δ`ILS`		Δ`Node`	
	`CQR`	`COQR`	`CQR`	`COQR`	`CQR`	`COQR`	`CQR`	`COQR`	`CQR`	`COQR`
facebook_1	.039 (.009)	**.016 (.003)**	1e-4 (1e-5)	**1e-5 (1e-6)**	5.187 (.408)	**3.379 (.410)**	10.395 (.376)	**3.348 (.168)**	2.945 (.534)	**2.078 (.334)**
facebook_2	.068 (.011)	**.016 (.003)**	1e-4 (1e-5)	**1e-5 (1e-5)**	5.175 (.342)	**3.648 (.317)**	8.625 (.278)	**4.026 (.153)**	3.538 (.427)	**1.682 (.238)**
blog_data	.018 (.003)	**.016 (.003)**	**1e-5 (1e-6)**	1e-5 (1e-5)	9.658 (.318)	**7.303 (.402)**	14.131 (.348)	**1.332 (.175)**	6.216 (.677)	**3.768 (.344)**
bio	.080 (.002)	**.039 (.003)**	1e-4 (1e-6)	**1e-5 (1e-6)**	**1.222 (.142)**	1.598 (.206)	7.565 (.286)	**4.601 (.114)**	**.757 (.120)**	1.064 (.179)
kin8nm	.231 (.003)	**.189 (.007)**	1e-4 (1e-5)	**1e-4 (1e-5)**	**1.708 (.265)**	2.896 (.375)	14.443 (.509)	**6.046 (.327)**	**1.833 (.322)**	1.859 (.218)
naval	.270 (.005)	**.061 (.007)**	.001 (1e-5)	**1e-4 (1e-5)**	**2.338 (.406)**	4.142 (.632)	8.718 (.557)	**2.769 (.332)**	**2.280 (.322)**	2.512 (.469)
meps_19	.122 (.007)	**.060 (.007)**	1e-4 (1e-5)	**1e-5 (1e-5)**	11.942 (.708)	**4.915 (.565)**	14.824 (.563)	**2.406 (.302)**	8.570 (1.345)	**5.413 (.969)**
meps_20	.104 (.006)	**.059 (.007)**	1e-4 (1e-5)	**1e-5 (1e-5)**	9.638 (.802)	**5.060 (.606)**	13.447 (.544)	**2.134 (.297)**	7.278 (1.430)	**4.084 (.754)**
meps_21	.118 (.008)	**.069 (.006)**	1e-4 (1e-5)	**1e-4 (1e-5)**	10.310 (.926)	**6.860 (.639)**	14.119 (.578)	**2.899 (.364)**	7.204 (1.294)	**4.515 (.858)**

Table S15: Real data: Average metric value (standard error) of a quantile regression forest model, fitted with 200 estimators and having at least 40 samples per leaf.

Dataset Name	`Coverage (%)`	`Length`	`corr`	`HSIC`	Δ`WSC`
facebook_1	96.349 (0.073)	1.132 (0.006)	0.049 (0.005)	3e-5 (3e-6)	1.325 (0.557)
blog_data	96.477 (0.005)	1.411 (0.003)	0.059 (0.002)	3e-5 (1e-5)	0.279 (0.174)
bio	93.797 (0.166)	2.173 (0.004)	0.043 (0.002)	1e-5 (1e-6)	1.508 (0.847)
kin8nm	93.887 (0.225)	2.475 (0.007)	0.033 (0.021)	2e-5 (9e-6)	2.389 (0.519)
naval	99.546 (0.015)	1.313 (0.004)	0.077 (0.009)	9e-6 (1e-6)	1.018 (0.215)
meps_19	95.745 (0.146)	1.218 (0.016)	0.049 (0.009)	1e-5 (1e-6)	0.753 (0.065)
meps_20	95.829 (0.076)	1.233 (0.023)	0.047 (0.015)	1e-5 (7e-6)	0.909 (0.211)
meps_21	95.668 (0.076)	1.221 (0.016)	0.047 (0.007)	2e-5 (6e-6)	0.89 (0.265)

[3] Physicochemical properties of protein tertiary structure data set. `https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure`. Accessed: January, 2019.

[4] Kinematics of an 8 link robot arm. `http://ftp.cs.toronto.edu/pub/neuron/delve/data/tarfiles/kin-family/`. Accessed: May, 2021.

[5] Condition based maintenance of naval propulsion plants data set. `http://archive.ics.uci.edu/ml/datasets/Condition+Based+Maintenance+of+Naval+Propulsion+Plants`. Accessed: May, 2021.

[6] Medical expenditure panel survey, panel 19. `https://meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-181`. Accessed: January, 2019.

[7] Medical expenditure panel survey, panel 20. `https://meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-181`. Accessed: January, 2019.

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

[9] Youngseog Chung, Willie Neiswanger, Ian Char, and Jeff Schneider. Beyond pinball loss: Quantile methods for calibrated uncertainty quantification. *arXiv preprint arXiv:2011.09588*, 2020.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[11] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The abstract and introduction reflect our contributions and the scope of the methods we offer.
   (b) Did you describe the limitations of your work? [Yes] We summarize the limitations in Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] In Section 6 we discuss how the proposed method can help alleviating biased predictions.

    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We read the ethics review guidelines and ensured that the paper conforms to them.

2. If you are including theoretical results...
    (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.
    (b) Did you include complete proofs of all theoretical results? [Yes] See Section S2 of the Supplementary Material.
3. If you ran experiments...
    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We add it to the supplemental material.
    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section S4 of the Supplementary Material.
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Supplementary Section S6
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The resources are described in Section S4.5 of the Supplementary Material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [Yes] We provide links to software packages we used to conduct our experiments. See Section 4 as well as Section S4 of the Supplementary Material.
    (b) Did you mention the license of the assets? [N/A] The data sets we used are publicly available.
    (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We do not add any new assets.
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The data sets we used are publicly available.
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data sets we used are publicly available.
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not use crowdsourcing or conducted research with human subjects.