A Proofs

A.1 Proof of Lemma 3.1

Proof. This is a well-known result; we prove it for completeness (see also Tibshirani et al. (2019) for an identical proof). Given support points $v_1, \ldots, v_n \in \mathbb{R}$ for a discrete distribution F, let $q = \text{Quantile}(\beta; F)$. Any points $v_i > q$ do not affect this quantile, i.e., if we consider a new distribution $\tilde{F}$ where all points $v_i > q$ are mapped to arbitrary values also larger than q. then $\text{Quantile}(\beta; F) = \text{Quantile}(\beta; \tilde{F})$. Accordingly, for the nonconformity scores V_i, we have that

$$V_{n+1} > \text{Quantile}(\beta; V_{1:n} \cup \{\infty\}) \iff V_{n+1} > \text{Quantile}(\beta; V_{1:(n+1)}).$$

Equivalently, we also have that

$$V_{n+1} \leq \text{Quantile}(\beta; V_{1:n} \cup \{\infty\}) \iff V_{n+1} \leq \text{Quantile}(\beta; V_{1:(n+1)}).$$

Given the discrete distribution over the $n+1$ V_i, $V_{n+1} \leq \text{Quantile}(\beta; V_{1:(n+1)})$ implies that V_{n+1} is among the $\lceil \beta(n+1) \rceil$ smallest of $V_{1:(n+1)}$. By exchangeability, this event occurs with probability at least $\frac{\lceil \beta(n+1) \rceil}{n+1} \geq \beta$. □

A.2 Proof of Theorem 3.2

Proof. This is also a well-known result; we prove it here for completeness (and see Tibshirani et al. (2019) for an identical proof). For notational convenience, let $V_i := V_i^{(X_{n+1}, Y_{n+1})}$. Y_{n+1} is included in $\mathcal{C}_\epsilon(X_{n+1})$ iff $V_{n+1} \leq \text{Quantile}(1-\epsilon; V_{1:n} \cup \{\infty\})$. As the nonconformity measure $\mathcal{S}$ preserves exchangeability by construction, if (X_i, Y_i) for $i = 1, \ldots, n+1$ are exchangeable, then so to are the nonconformity scores V_i, $i = 1, \ldots, n+1$. We can then apply Lemma 3.1 to complete the proof. □

A.3 Proof of Lemma 4.2

Proof. Let the event $\{T_i = t_i\}$ indicate that task i has a quantile prediction $\{\widehat{Q}_i = q_i)$ and distribution function $\{F_i = f_i\}$ over meta nonconformity scores given $\widehat{\mathcal{S}}$.

For notational convenience, assume tasks T_i, $i \in \mathcal{I}_{\text{cal}}$ and T_{t+1} are indexed contiguously as $i = 1, \ldots, n+1$. Next, denote by E_t the event that $\{T_1, \ldots, T_{n+1}\} = \{t_1, \ldots, t_{n+1}\}$, i.e., we observe an unordered set of task values. Exchangeability of tasks T_i implies that

$$\mathbb{P}(T_{n+1} = t_i \mid E_t) = \frac{1}{n+1},$$

and, accordingly, that the distribution of $T_{n+1} \mid E_t$ is uniform on the set $\{t_1, \ldots, t_{n+1}\}$.

Again for notational convenience, let

$$\widehat{V}_i := \widehat{V}_{i,k+1}^{(X_i^{\text{test}}, Y_i^{\text{test}})}$$

i.e., we use $\widehat{V}_i$ to denote the meta nonconformity score for task i's random test point.

For any scalar $\lambda \in \mathbb{R}$, we can then write

$$\begin{aligned}
&\mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid E_t) \\
&= \sum_{i=1}^{n+1} \mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \lambda, T_{n+1} = t_i \mid E_t) \\
&= \sum_{i=1}^{n+1} \mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid T_{n+1} = t_i) \mathbb{P}(T_{n+1} = t_i \mid E_t) \\
&= \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid T_{n+1} = t_i).
\end{aligned}$$

Since the event $\{T_{n+1} = t_i\}$ implies $\{\widehat{Q}_{n+1} = q_i, F_{n+1} = f_i\}$, we can reduce this to

$$\mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid E_t) = \frac{1}{n+1} \sum_{i=1}^{n+1} f_i(q_i + \lambda).$$

Furthermore, on the event E_t, we have $\{T_1, \ldots, T_{n+1}\} = \{t_1, \ldots, t_{n+1}\}$, so (with slight abuse of notation)

$$\mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid E_t) = \frac{1}{n+1} \sum_{i=1}^{n+1} F_i(\widehat{Q}_i + \lambda).$$

As F_i is a distribution function with range $[0, 1]$, we can remove T_{n+1} from the summation to get a lower bound,

$$\mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid E_t) \geq \frac{1}{n+1} \sum_{i=1}^{n} F_i(\widehat{Q}_i + \lambda).$$

For a fixed β, substitute $\Lambda(\beta; \mathcal{I}_{\text{cal}})$ for λ to derive

$$\begin{aligned}
&\mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \Lambda(\beta; \mathcal{I}_{\text{cal}}) \mid E_t) \\
&\qquad \geq \frac{1}{n+1} \sum_{i=1}^{n} F_i(\widehat{Q}_i + \Lambda(\beta; \mathcal{I}_{\text{cal}})) \geq \beta.
\end{aligned}$$

Because this is true for any E_t, we can marginalize to obtain

$$\begin{aligned}
&\mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \Lambda(\beta; \mathcal{I}_{\text{cal}})) \\
&\quad = \int_{E_t} \mathbb{P}(\widehat{V}_{n+1} \leq \widehat{Q}_{n+1} + \Lambda(\beta; \mathcal{I}_{\text{cal}}) \mid E_t) \, d\mathbb{P}(E_t) \\
&\quad \geq \beta \int_{E_t} d\mathbb{P}(E_t) = \beta.
\end{aligned}$$

□

Symbol	Meaning
k	The number of in-task examples used for few-shot learning.
ϵ	The stipulated performance tolerance.
δ	The stipulated secondary confidence tolerance for calibration conditional validity.
$\mathcal{T}$	The space of potential tasks to be solved in a few-shot learning setting.
$\mathcal{X} \times \mathcal{Y}$	The joint input (X's) and output (Y's) space.
$\mathcal{I}_{\text{train}}$	The set of auxiliary tasks used for meta-learning nonconformity scores and quantile predictors.
$\mathcal{I}_{\text{cal}}$	The set of auxiliary tasks used to calibrate the quantile predictor.
T_{t+1}	The target few-shot test task to be solved.
$\widehat{\mathcal{S}}$	A meta-learned nonconformity measure.
$\widehat{\mathcal{P}}_{1-\epsilon}$	A meta-learned regressor of the $1-\epsilon$ quantile of $\widehat{\mathcal{S}}$'s scores on T_{t+1} given k in-task samples.
$\widehat{V}_{i,j}^{(x,y)}$	The meta nonconformity score for example j of task i, given the current candidate output (x, y).
$F_i, \widehat{F}_{m_i}$	The true vs. m_i-sample empirical distribution function over nonconformity scores of task i.
$\widehat{Q}_i$, $\text{Quantile}(1-\epsilon, F_i)$	The predicted vs. true nonconformity score $1-\epsilon$ quantiles for task i.
$\Lambda(1-\epsilon, \mathcal{I}_{\text{cal}})$	The $1-\epsilon$ meta quantile correction factor, computed using calibration tasks.
$\mathcal{C}_\epsilon$, $\mathcal{M}_\epsilon$	Output label sets for standard and meta conformal prediction, respectively, at level $1-\epsilon$.

Table A.1. Definitions of selected common notations used in this paper.

A.4 Proof of Theorem 4.3

Proof. Again, for notational convenience, let

$$\widehat{V}_i := \widehat{V}_{i,k+1}^{(X_i^{\text{test}}, Y_i^{\text{test}})}$$

Y_{t+1}^{test} is included in $\mathcal{M}_\epsilon(X_{t+1}^{\text{test}})$ iff $\widehat{V}_{t+1} \leq \widehat{Q}_{t+1} + \Lambda(1-\epsilon; \mathcal{I}_{\text{cal}})$. As $\widehat{\mathcal{S}}$ and $\widehat{\mathcal{P}}$ are trained on the disjoint proper training set $\mathcal{I}_{\text{train}}$, they preserve exchangeability, and produce exchangeable $\widehat{Q}_i$. We can then apply Lemma 4.2. □

A.5 Proof of Proposition 4.5

Proof. Again, for notational convenience, let

$$\widehat{V}_i := \widehat{V}_{i,k+1}^{(X_i^{\text{test}}, Y_i^{\text{test}})}.$$

As stated in the claim, assume that as $k \to \infty$,

$$\left|\widehat{\mathcal{P}}_{1-\epsilon}(Z_{i,1:k}; \phi_{\text{meta}}) - \text{Quantile}(1-\epsilon, F_i)\right| = o_{\mathbb{P}}(1),$$

where F_i is the distribution of $\widehat{V}_i$. That is, the quantile converges in probability to the true quantile where $\forall \alpha, \mu$ there exists $K_{\alpha,\mu}$ such that

$$\mathbb{P}\left(\left|\widehat{\mathcal{P}}_{1-\epsilon}(Z_{i,1:k}; \phi_{\text{meta}}) - \text{Quantile}(1-\epsilon, F_i)\right| \geq \mu\right) \leq \alpha,$$

$\forall k > K_{\alpha,\mu}$. This is a standard property of consistent estimators (e.g., see Lei et al. (2018) for similar assumptions).

As $\Lambda(1-\epsilon; \mathcal{I}_{\text{cal}}) \geq 0$, for any target task $t_{t+1} \in \mathcal{T}$, we have that the corrected quantile, $\widetilde{Q}_{t+1} = \widehat{Q}_{t+1} + \Lambda(1-\epsilon; \mathcal{I}_{\text{cal}})$, is always *conservative* for large enough k, i.e.,

$$\mathbf{1}\left\{\widetilde{Q}_{t+1} \geq \text{Quantile}(1-\epsilon, F_{t+1}) \mid T_{t+1} = t_{t+1}\right\} = 1 - o_{\mathbb{P}}(1). \tag{10}$$

In other words, this is to say that if $\widehat{\mathcal{P}}_{1-\epsilon}$ converges in probability to the true quantile, then $\widehat{\mathcal{P}}_{1-\epsilon}$ + some nonzero factor converges in probability to at least the true quantile.

Next, if $\widehat{V}_{t+1} \leq \widetilde{Q}_{t+1}$, then Y_{t+1}^{test} is included in $\mathcal{M}_\epsilon(X_{t+1}^{\text{test}})$ (according to the definition of $\mathcal{M}_\epsilon$). Furthermore, by the definition of Quantile, the event that $\widehat{V}_{t+1} \leq \text{Quantile}(1-\epsilon, F_{t+1})$ happens with probability at least $1-\epsilon$. Therefore, if $\widetilde{Q}_{t+1} \geq \text{Quantile}(1-\epsilon, F_{t+1})$, then Y_{t+1}^{test} is included in $\mathcal{M}_\epsilon(X_{t+1}^{\text{test}})$ with probability at least $1-\epsilon$. Combining with Eq. (10) completes our proof. □

A.6 Proof of Proposition 4.7

Proof. Let $\widehat{F}_{m_i}$ be the m_i-sample ECDF for T_i. Define the empirical correction, $\Lambda'(\beta, \mathcal{I}_{\text{cal}})$, when plugging in $\widehat{F}_{m_i}$ as

$$\inf\left\{\lambda \colon \frac{1}{|\mathcal{I}_{\text{cal}}|+1} \sum_{i \in \mathcal{I}_{\text{cal}}} \widehat{F}_{m_i}(\widehat{V}_{i,k+1}^{\text{test}} \leq \widehat{Q}_i + \lambda) \geq \beta\right\} \tag{11}$$

where the ECDF is calculated as

$$\widehat{F}_{m_i} := \sum_{j=1}^{m_i} \mathbf{1}\{\widehat{V}_{i,k+1}^{(j)} \leq \widehat{Q}_i + \lambda\},$$

where $\widehat{V}_{i,k+1}^{(j)}$ are i.i.d. and $\widehat{V}_{i,k+1}^{(j)} \overset{d}{=} \widehat{V}_{i,k+1}^{\text{test}}$.

We now proceed in two parts. First, we prove that if the approximation error incurred by using $\Lambda'(\beta, \mathcal{I}_{\text{cal}})$ is bounded by τ with probability $1-\delta$, then $\mathcal{M}_{\epsilon-\tau}$ is (δ, ϵ) valid. Second, we prove that the error is bounded according to Eq. (9).

(1) Following the proof of Lemma 4.2, we have that

$$\mathbb{P}(\widehat{V}_{t+1} \leq \widehat{Q}_{t+1} + \Lambda'(\beta; \mathcal{I}_{\text{cal}}) \mid E_t) \geq \frac{1}{|\mathcal{I}_{\text{cal}}|+1} \sum_{i \in \mathcal{I}_{\text{cal}}} F_i(\widehat{Q}_i + \Lambda'(\beta; \mathcal{I}_{\text{cal}})). \tag{12}$$

For ease of notation, let

$$A := \frac{1}{|\mathcal{I}_{\text{cal}}|+1} \sum_{i \in \mathcal{I}_{\text{cal}}} \widehat{F}_{m_i}(\widehat{Q}_i + \Lambda'(\beta; \mathcal{I}_{\text{cal}})),$$
$$B := \frac{1}{|\mathcal{I}_{\text{cal}}|+1} \sum_{i \in \mathcal{I}_{\text{cal}}} F_i(\widehat{Q}_i + \Lambda'(\beta; \mathcal{I}_{\text{cal}})).$$

Next, assume that (to be proved) for some $\tau > 0$

$$\mathbb{P}(A - B < \tau) \geq 1 - \delta. \tag{13}$$

By construction—see Eq. (11)—we have $A \geq \beta$. Then by Eq. (13), we have that with probability $1 - \delta$,

$$B > A - \tau \geq \beta - \tau.$$

Choose $\beta \geq 1 - \epsilon + \tau$. Then $B \geq 1 - \epsilon$. By convention, this corresponds to $\beta := 1 - \epsilon' \geq 1 - (\epsilon - \tau)$, or $\epsilon' \leq \epsilon - \tau$ as in Eq. (9). Combining this with Eq. (12), we have

$$\mathbb{P}(\widehat{V}_{t+1} \leq \widehat{Q}_{t+1} + \Lambda'(\beta; \mathcal{I}_{\text{cal}}) \mid E_t) \geq 1 - \epsilon.$$

This is true for all E_t, so we can marginalize to obtain

$$\mathbb{P}(\widehat{V}_{t+1} \leq \widehat{Q}_{t+1} + \Lambda'(\beta; \mathcal{I}_{\text{cal}})) \geq 1 - \epsilon.$$

(2) We now prove the assumption stated in Eq. (13). Given an m-sample ECDF, $\widehat{F}_m(u)$, for some random variable U, the Dvoretsky-Kiefer-Wolfowitz inequality allows us to build a confidence interval for the value of the true distribution function, $F(u)$, where

$$\mathbb{P}\left(\sup_{u \in \mathbb{R}} |\widehat{F}_m(u) - F(u)| > \gamma\right) \leq 2e^{-2n\gamma^2}.$$

Alternatively stated, with probability at least $1 - \alpha$, $F(u) \in [\widehat{F}_m(u) - \gamma, \widehat{F}_m(u) + \gamma]$, where $\gamma = \sqrt{\frac{\log \frac{2}{\alpha}}{2n}}$.

We combine this result with Hoeffding's inequality.

Let $Y_i := \widehat{F}_{m_i}(\widehat{V}_i \leq \widehat{Q}_i + \lambda) - F_i(\widehat{V}_i \leq \widehat{Q}_i + \lambda)$. Once again for notational convenience, assume tasks T_i, $i \in \mathcal{I}_{\text{cal}}$ and T_{t+1} are indexed contiguously as $i = 1, \ldots, n+1$. The difference, $A - B$, is then equivalent to $\frac{1}{n+1}\sum_{i=1}^n Y_i$. According to our assumptions, Y_i's are i.i.d., $\mathbb{E}[Y_i] = \mathbb{E}[\widehat{F}_{m_i}] - \mathbb{E}[F_i] = 0$, and $Y_i \in [-\gamma_i, \gamma_i]$ w.p. $1 - \alpha$. As above, we define $\gamma_i = \sqrt{\frac{\log \frac{2}{\alpha}}{2m_i}}$.

Applying Hoeffding's inequality gives

$$\begin{aligned}
&\mathbb{P}\Big(\frac{1}{n+1}\sum_{i=1}^n Y_i < \tau\Big) \\
&\geq \mathbb{P}\Big(\sum_{i=1}^n Y_i < n\tau, \bigcap_{i=1}^n Y_i \in [-\gamma_i, \gamma_i]\Big) \\
&\geq \mathbb{P}\Big(\sum_{i=1}^n Y_i < n\tau \Big| \bigcap_{i=1}^n Y_i \in [-\gamma_i, \gamma_i]\Big)\mathbb{P}\Big(\bigcap_{i=1}^n Y_i \in [-\gamma_i, \gamma_i]\Big) \\
&\geq \Big(1 - e^{-\frac{2n^2\tau^2}{\sum_{i=1}^n (2\gamma_i)^2}}\Big)\Big(1 - \alpha\Big)^n.
\end{aligned}$$

Solving for τ given the target $1 - \delta$ error probability yields

$$1 - \delta = \Big(1 - e^{-\frac{2n^2\tau^2}{\sum_{i=1}^n (2\gamma_i)^2}}\Big)\Big(1 - \alpha\Big)^n$$
$$\tau = \sqrt{\frac{-2}{n^2}\Big(\sum_{i=1}^n \gamma_i^2\Big) \log\Big(1 - \frac{1-\delta}{(1-\alpha)^n}\Big)}$$

This is valid for any choice of α (as long as the log term is defined), so we are free to choose α that minimizes τ. □

B Meta Conformal Prediction Details

B.1 Meta-Learning algorithms

Prototypical networks (Snell et al., 2017). We use prototypical networks for our classification tasks. We assume that for each task we have N total classes with K examples per class (for a total of $k = N \times K$ training examples). In this model, an encoder, $\mathbf{h} = \text{enc}(x; \theta)$ is trained to produce vector representations. Thereafter, a "prototype" for each class is computed by averaging the representations of all instances of that class. Let S_j denote the support set of training examples for class j. Then the prototype $\mathbf{c}_j$ is

$$\mathbf{c}_j := \frac{1}{|S_j|} \sum_{(x_i, y_i) \in S_j} \text{enc}(x_i; \theta).$$

The likelihood of each class is then calculated using a softmax over the euclidean distance to each prototype:

$$p_\theta(y = j \mid x) := \frac{\exp(-d(\mathbf{c}_j, \text{enc}(x; \theta)))}{\sum_{j'} \exp(-d(\mathbf{c}_{j'}, \text{enc}(x; \theta)))}, \tag{14}$$

where $d(\cdot, \cdot)$ denotes the euclidean distance.

During training, random training "episodes" are created by sampling N classes from the training set. For each class, K examples are randomly sampled to construct the prototypes. An additional Q examples are then sampled to simulate queries. The optimization objective is to then minimize the cross entropy loss across queries.

After training, we use $-p_\theta(y = j \mid x)$ as defined in Eq. (14) as the nonconformity measure for label $y = j$.

Differentiable ridge regression (Bertinetto et al., 2019). We use differentiable ridge regression networks for our regression tasks. We assume that for each task we have k labeled (x_i, y_i) pairs, where $y \in \mathbb{R}$. In this model, like the prototypical networks, an encoder, $\mathbf{h} = \text{enc}(x; \theta)$, is trained to produce vector representations of dimension d. We then solve a least-squares regression to obtain our prediction, $\hat{y} = \mathbf{w} \cdot \text{enc}(x; \theta)$, where

$$\mathbf{w} = X^\top (XX^\top + \lambda I)^{-1} Y$$

with $X \in \mathbb{R}^{k \times d}$, $Y \in \mathbb{R}^k$, and λ a meta regularization parameter that we optimize. We optimize MSE by back-propagating through the least-squares operator to the encoder. We train using the same episode-based procedure that we described for the prototypical networks.

After training, we use the absolute error, $|\hat{y} - y|$, as the nonconformity score for candidate $y \in \mathbb{R}$.

Deep sets (Zaheer et al., 2017). We use a simple deep sets architecture for all of our quantile predictors. Deep sets are of the form

$$f(X) := \text{dec}\Big(\sum_{x \in X} \text{enc}(x; \phi_1); \phi_2\Big)$$

where X is an input set of elements, enc is an element-wise encoder, and dec is a decoder that operates on the aggregated encoded set elements. Importantly, the deep sets model f is invariant to permutations of the elements in X.

B.2 Implementation details

Image classification. Each image is first resized to 84×84 pixels. We use a CNN encoder with 4 layers. Each layer contains a 3×3 convolution kernel with 64 channels and a padding of size 1, followed by batch normalization layer, ReLU activation, and a 2×2 max pooling filter. The final output is of size 1600, which we use to compute the prototypes and as the query representations. We train the model for 100 epochs with an Adam optimizer and a batch size of 256. In each epoch, we run 100 episodes in which we sample 10 support images and 15 query images per class.

Relation classification. We use GloVe (Pennington et al., 2014) word embeddings of size 50 to convert the sentence into vectors. To each word embedding, we also concatenate two learned position embeddings of size 5, where the positions are relative to the location of the two entities in the sentence. Thereafter, a 1D convolution is applied with 230 output channels, a kernel size of 3 and padding size 1, followed by a ReLU activation. Finally, a max pooling filter is applied. The resultant sentence representation of size 230 is used to compute the prototypes and query representations. We train the model for a total of $20k$ episodes with a SGD optimizer and a batch size of 32. In each episode, we sample 10 support sentences and 5 query sentences per class.

Chemical property prediction. Our ridge regression network uses directed message passing networks (Yang et al., 2019) to compute $\text{enc}(x; \theta)$. The message passing network uses graph convolutions to learn a deep molecular representation that is shared across property predictions. We also include additional RDKit features as inputs.[6] We map inputs with a FFNN with hidden size 200, and then apply 3 layers of graph convolutions with a hidden size of 256. Finally, we map the output representation to a hidden size of 16, and apply least-squares regression. We train the network using an Adam optimizer for 15 epochs with 8 meta episodes per batch, each with 32 queries (for a total batch size of 256).

Quantile prediction. For all of our quantile predictors, we use a 2-layer FFNN for both the element-wise encoder, $\text{enc}(\cdot; \phi_1)$, and the aggregated set decoder, $\text{dec}(\cdot; \phi_2)$. Each FFNN has a hidden size of 256 and uses ReLU activations. We train the network using an Adam optimizer for 15 epochs with batch size 64.

B.3 Training strategy

We adopt a cross-fold procedure for training our meta nonconformity measure $\widehat{\mathcal{S}}$ and meta quantile predictor $\widehat{\mathcal{P}}_{1-\epsilon}$ in a data efficient way, as outlined in §4.2. Figure B.1 illustrates this cross-fold process, in which we train a meta-nonconformity measure on each training fold and aggregate their predictions as input data for the quantile predictor.

Since we train in a cross-fold manner but ultimately use a meta-nonconformity measure $\widehat{\mathcal{S}}$ that is trained on *all* of the training data, there is a train-test mismatch in the data supplied to the quantile predictor. Nevertheless, any error induced by this discrepancy (and any other sources of error, for that matter) is handled during meta-calibration (§4.3).

All experiments took 1-5 hours to run on an Nvidia 2080 Ti GPU. As absolute performance is not the primary goal of this work, little hyperparameter tuning was done (most hyperparameters were taken from prior work). Datasets are available for *mini*ImageNet[7], FewRel 1.0[8], and ChEMBL[9].

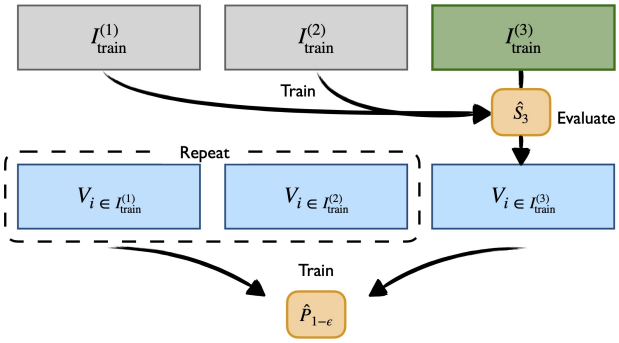

Figure B.1. An illustration of our strategy for learning meta nonconformity measures, $\widehat{\mathcal{S}}$, and meta quantile predictors, $\widehat{\mathcal{P}}_{1-\epsilon}$. As $\widehat{\mathcal{P}}_{1-\epsilon}$ is trained on the outputs of $\widehat{\mathcal{S}}$, we adopt a cross-fold procedure where we first train $\widehat{\mathcal{S}}$ on a fraction of the data, and evaluate nonconformity scores on the held-out fold. We repeat this process for all k_f folds, and then aggregate them all for training $\widehat{\mathcal{P}}_{1-\epsilon}$.

[6] www.rdkit.org

[7] https://github.com/yaoyao-liu/mini-imagenet-tools

[8] https://thunlp.github.io/1/fewrel1.html

[9] https://github.com/chemprop/chemprop