Sample Compression Hypernetworks: From Generalization Bounds to Meta-Learning

Benjamin Leblanc, Mathieu Bazinet, Nathaniel D'Amours
Université Laval
{benjamin.leblanc.2, mathieu.bazinet.2, nathaniel.damours.1}@ulaval.ca

Alexandre Drouin
ServiceNow Research, Université Laval
alexandre.drouin@servicenow.com

Pascal Germain
Université Laval, Canada CIFAR AI Chair
pascal.germain@ift.ulaval.ca

Abstract

Reconstruction functions are pivotal in sample compression theory, a framework for deriving tight generalization bounds. From a small sample of the training set (the compression set) and an optional stream of information (the message), they recover a predictor previously learned from the whole training set. While usually fixed, we propose to learn reconstruction functions. To facilitate the optimization and increase the expressiveness of the message, we derive a new sample compression generalization bound for real-valued messages. From this theoretical analysis, we then present a new hypernetwork architecture that outputs predictors with tight generalization guarantees when trained using an original meta-learning framework. The results of promising preliminary experiments are then reported.

1 Introduction

Initiated by Littlestone and Warmuth [20] and refined by many authors [1, 3, 4, 5, 6, 10, 11, 12, 13, 19, 22, 23, 25, 28, 30], the sample compression theory expresses generalization bounds on predictors that rely only on a small subset of the training set, referred to as the *compression set*. The provided statistical guarantees are valid even if the learning algorithm observes the entire training dataset, as long as there exists a *reconstruction function* that recovers the learned predictor from the compression set and, optionally, a short stream of additional information (referred to as the *message*). The sample compression theorems thus express the generalization ability of predictive models as an accuracy-complexity trade-off, measured respectively by the training loss and the size of the compressed representation, which has been the motivation for unconventional yet successful learning algorithms.

Among sample compress learning algorithms, a first line of work that led to practical machine learning algorithms was pioneered by Marchand and Shawe-Taylor [21, 22] and their *Set Covering Machine* (SCM) learning algorithm, a greedy iterative procedure that selects a very small subset of the training set to build a decision rule based on data-dependent features. A second line of work is rooted in the theoretical work of Campi et al. [4] and is incarnated by the *Pick-to-learn* meta-algorithm recently proposed by Paccagnan et al. [26]. Such sample compression learning algorithms are typically expressed as a discrete optimization procedure tailored for a well-specified reconstruction function.

The originality of our contribution lies in the *learning of the reconstruction function*, which is achieved by making the reconstruction function a direct component of our learning algorithm. The resulting architecture can be viewed as a new form of encoder-decoder that *compresses* a dataset into a compression set and a message, and *reconstructs* a predictor. We leverage the proposed autoencoder in

38th Conference on Neural Information Processing Systems (NeurIPS 2024) - ML and Compression Workshop.

a meta-learning framework, leading to tight task-specific sample compression generalization bounds. This is achievable thanks to an original sample compression theorem for real-valued messages.

2 The Sample Compression Setting

The prediction problem. A dataset $S = \{z_j\}_{j=1}^m$ is a collection of m examples, each of them being a feature-target pair $z = (\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, and a predictor is a function $h : \mathcal{X} \to \mathcal{Y}$. We denote $\mathcal{H}$ as the predictor space. Let A be a learning algorithm $A : \bigcup_{k\in\mathbb{N}} (\mathcal{X} \times \mathcal{Y})^k \to \mathcal{H}$ that outputs a predictor $A(S) \in \mathcal{H}$. Given a predictor h and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$, the empirical loss of the predictor over a set of m independently and identically distributed (*i.i.d.*) examples is $\widehat{\mathcal{L}}_S(h) = \frac{1}{m}\sum_{j=1}^m \ell(h(\mathbf{x}_j), y_j)$. We denote $\mathcal{D}$ the data-generating distribution over $\mathcal{X} \times \mathcal{Y}$ such that $S \sim \mathcal{D}^n$ and the generalization loss of a predictor h is $\mathcal{L}_\mathcal{D}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} [\ell(h(\mathbf{x}), y)]$.

The reconstruction function. Once a predictor h is learned from a dataset S, *i.e.* $h = A(S)$, one can obtain an upper bound on $\mathcal{L}_\mathcal{D}(h)$ thanks to the sample compression theory whenever it is possible to *reconstruct* the predictor h from a compression set (that is, a subset of S) and an optional message (chosen from a predetermined discrete messages set Σ). This is performed by a reconstruction function, $\mathcal{R} : \bigcup_{k\in\mathbb{N}} (\mathcal{X} \times \mathcal{Y})^k \times \Sigma \to \mathcal{H}$. Thus, a sample compression predictor can be written $h = \mathcal{R}(S_\mathbf{j}, \sigma)$, with $\mathbf{j} \subset \{i\}_{i=1}^m$ being the indexes of the training samples belonging to the compression set $S_\mathbf{j} = \{z_j\}_{j\in\mathbf{j}}$, and $\sigma \in \Sigma$ being the message. In the following, we denote the set of all training indices $\mathbf{m} = \{i\}_{i=1}^m$, and $\mathcal{P}(\mathbf{m})$ its powerset; for compression set indices $\mathbf{j} \in \mathcal{P}(\mathbf{m})$, the complement is $\bar{\mathbf{j}} = \mathbf{m} \setminus \mathbf{j}$.

Notable theoretical results. Theorem 1 below, due to Marchand and Sokolova [23], improves the bound developed for the SCM algorithm [21, 22]. It is premised on two data-independent distributions: $P_{\mathcal{P}(\mathbf{m})}$ on the compression set indices $\mathcal{P}(\mathbf{m})$, and P_Σ on a discrete set of messages Σ. Noteworthy, the bound is valid solely for the zero-one loss, as it considers each "successful" and "unsuccessful" prediction to be the result of a Bernoulli distribution.

Theorem 1 (Sample compression - binary loss with discrete messages [23]). *For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, for any set $J \subseteq \mathcal{P}(\mathbf{m})$, for any distribution P_J over J, for any P_Σ over Σ, for any reconstruction function $\mathcal{R}$, for any binary loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$ and for any $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the draw of $S \sim \mathcal{D}^m$, we have*

$$\forall \mathbf{j} \in J, \sigma \in \Sigma \ :$$

$$\mathcal{L}_\mathcal{D}\big(\mathcal{R}(S_\mathbf{j}, \sigma)\big) \leq \underset{r\in[0,1]}{\operatorname{argsup}} \left\{ \sum_{k=1}^{K} \binom{|\bar{\mathbf{j}}|}{k} r^k (1-r)^{|\bar{\mathbf{j}}|-k} \geq P_J(\mathbf{j})\, P_\Sigma(\sigma)\, \delta \;\middle|\; K = |\bar{\mathbf{j}}| \widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}\big(\mathcal{R}(S_\mathbf{j}, \sigma)\big) \right\}.$$

Theorem 1 is limited in its scope, for many tasks involve non-binary loss (e.g. regression tasks, or classification where making a given error has a bigger impact than others). The following recent result [3] permits real-valued losses $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Given a *comparator function* $\Delta : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, it bounds the discrepancy between the empirical loss of the reconstructed hypothesis $\mathcal{R}(S_\mathbf{j}, \sigma)$ on the complement set $S_\mathbf{j}$ and the generalization loss on the data distribution $\mathcal{D}$.

Theorem 2 (Sample compression - real-valued losses with discrete messages [3]). *For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, for any set $J \subseteq \mathcal{P}(\mathbf{m})$, for any distribution P_J over J, for any distribution P_Σ over Σ, for any reconstruction function $\mathcal{R}$, for any loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, for any function $\Delta : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and for any $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the draw of $S \sim \mathcal{D}^m$, we have:*

$$\forall \mathbf{j} \in J, \sigma \in \Sigma : \Delta\Big(\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_\mathbf{j}, \sigma)), \mathcal{L}_\mathcal{D}(\mathcal{R}(S_\mathbf{j}, \sigma))\Big) \leq \frac{1}{m - |\mathbf{j}|}\left[\ln\left(\frac{\mathcal{E}_\Delta(\mathbf{j}, \sigma)}{P_J(\mathbf{j}) \cdot P_\Sigma(\sigma) \cdot \delta}\right)\right],$$

with

$$\mathcal{E}_\Delta(\mathbf{j}, \sigma) = \underset{T_\mathbf{j}\sim\mathcal{D}^{|\mathbf{j}|}}{\mathbb{E}} \ \underset{T_{\bar{\mathbf{j}}}\sim\mathcal{D}^{m-\bar{\mathbf{j}}}}{\mathbb{E}} e^{|\bar{\mathbf{j}}|\Delta\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{j}}}}(\mathcal{R}(T_\mathbf{j},\sigma)), \mathcal{L}_\mathcal{D}(\mathcal{R}(T_\mathbf{j},\sigma))\right)}.$$

In order to compute a numerical bound on the generalization loss $\mathcal{L}_\mathcal{D}(\mathcal{R}(S_\mathbf{j}, \sigma))$, one must commit to a choice of Δ. See Appendix A for corollaries involving specific choices of comparator function.

3 A New Sample Compression Bound for Continuous Messages

Our first contribution lies in the extension of Theorem 2 to real-valued messages, to both ease the optimization of the proposed architecture by back-propagation and allow for a more complex message space. This is achieved by using a strategy from the PAC-Bayesian theory [24]: we consider a data-independent prior distribution over the messages Σ, denoted P_Σ, and a data-dependent posterior distribution, denoted Q_Σ, over the messages. We then obtain a bound for the expected loss over Q_Σ.

Theorem 3 (Sample compression - real-valued losses with continuous messages)**.** *For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, for any set $J \subseteq \mathcal{P}(\mathbf{m})$ such that $\max_{\mathbf{j} \in J} |\mathbf{j}| = c$, for any distribution P_J over J, for any prior distribution P_Σ over Σ, for any reconstruction function $\mathcal{R}$, for any loss $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$, for any convex function $\Delta : [0,1] \times [0,1] \to \mathbb{R}$ and for any $\delta \in (0,1]$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^m$, we have:*

$$\forall \mathbf{j} \in J, Q_\Sigma \text{ over } \Sigma :$$

$$\Delta\left(\mathop{\mathbb{E}}_{\sigma \sim Q_\Sigma} \widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_\mathbf{j}, \sigma)), \mathop{\mathbb{E}}_{\sigma \sim Q_\Sigma} \mathcal{L}_\mathcal{D}(\mathcal{R}(S_\mathbf{j}, \sigma)) \right) \leq \frac{1}{m-c}\left[\mathrm{KL}(Q_\Sigma || P_\Sigma) + \ln\left(\frac{\mathcal{J}_\Delta(m-c)}{P_J(\mathbf{j}) \cdot \delta} \right) \right],$$

with

$$\mathcal{J}_\Delta(m-c) = \mathop{\mathbb{E}}_{\sigma \sim P_\Sigma} \mathop{\mathbb{E}}_{T_\mathbf{j} \sim \mathcal{D}^{|\mathbf{j}|}} \mathop{\mathbb{E}}_{T_{\bar{\mathbf{j}}} \sim \mathcal{D}^{m-|\mathbf{j}|}} e^{(m-c)\cdot \Delta\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{j}}}}(\mathcal{R}(T_\mathbf{j}, \sigma)), \mathcal{L}_\mathcal{D}(\mathcal{R}(T_\mathbf{j}, \sigma)) \right)}.$$

See Appendix B for the complete proof of Theorem 3 and Appendix C for corollaries involving specific choices of Δ. The complexity term increases with the Kullback-Leibler divergence between the prior and the posterior, defined as $\mathrm{KL}(Q_\Sigma || P_\Sigma) = \mathbb{E}_{\sigma \sim Q_\Sigma} \log[Q_\Sigma(\sigma)/P_\Sigma(\sigma)]$. This new result shares similarities with the existing PAC-Bayes sample compression theory [8, 9, 17, 18], which gives PAC-Bayesian bounds for an expectation of data-dependent predictors given distributions on both the compression set and the messages. Our result differs by restricting the expectation solely according to the message.

4 Sample Compression Hypernetworks

The three sample compression theorems of the previous section assume a fixed reconstruction function $\mathcal{R}$. Instead, we propose learning it as a neural network $\mathcal{R}_\theta$ with parameters θ. This *reconstruction hypernetwork* $\mathcal{R}_\theta$ takes two complementary inputs:

1. A compression set $S_\mathbf{j}$ containing a fixed number c examples;
2. A message $\boldsymbol{\sigma}$ taking the form of a vector of fixed size b. We experiment with either real-valued messages ($\boldsymbol{\sigma} \in [-1,1]^b$), or discrete message ($\boldsymbol{\sigma} \in \{-1,1\}^b$).

The output of the reconstruction hypernetwork is an array $\gamma \in \mathbb{R}^{|\gamma|}$ that is in turn the parameters of a *downstream network* $h_\gamma : \mathbb{R}^d \to \mathcal{Y}$. Hence, given a training set S, a compression set $S_\mathbf{j} \subset S$ and a message $\boldsymbol{\sigma} \in \Sigma$ (the choice of $S_\mathbf{j}$ and $\boldsymbol{\sigma}$ is discussed in the next section), we train the reconstruction hypernetwork by optimizing its parameters θ in order to minimize the empirical loss of the downstream predictor h_γ on the complement set $S_{\bar{\mathbf{j}}} = S \setminus S_\mathbf{j}$:

$$\min_\theta \left\{ \frac{1}{m-|\mathbf{j}|} \sum_{(\mathbf{x},y)\in S_{\bar{\mathbf{j}}}} \ell\big(h_\gamma(\mathbf{x}), y\big) \;\middle|\; \gamma = \mathcal{R}_\theta(S_\mathbf{j}, \boldsymbol{\sigma}) \right\}. \tag{1}$$

Note that the above corresponds to the minimization of the empirical loss term $\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\cdot)$ of the sample compression bounds. However, to be statistically valid, these bounds must not be computed on the same data used to learn the reconstruction function. The next section describes a meta-learning framework that enables the use of the reconstruction hypernetwork to obtain generalization guarantees based on sample compression theory.

5 Combining Sample Compression and Meta-Learning

In the following, we extend our framework to the meta-learning analysis pioneered by Baxter [2], where a learning problem encompasses multiple *tasks*.

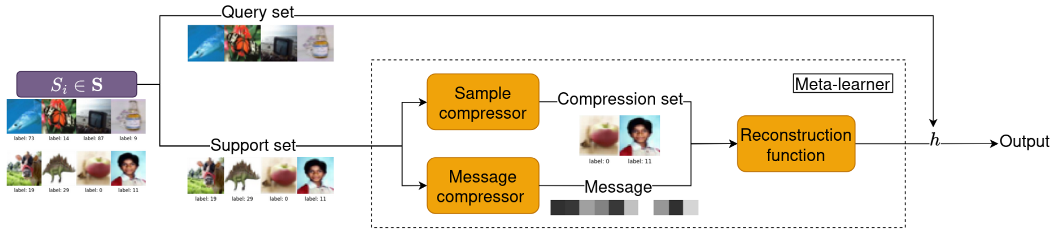

Figure 1: The proposed meta-learning framework.

The meta-prediction problem. Each *task* $\mathcal{D}_i$ is a realization of a meta distribution $\mathbf{D}$, and $S_i{\sim}\mathcal{D}_i^{m_i}$ contains m_i *i.i.d.* samples from a given task. A meta-learning algorithm is given a meta-dataset $\mathbf{S} = \{S_i\}_{i=1}^n$, that is a collection of n datasets obtained from distributions $\{\mathcal{D}_i\}_{i=1}^n$. The aim is to exploit the information in $\mathbf{S}$ so that, given only a few sample $S' \sim (\mathcal{D}')^{|S'|}$ from a new task $\mathcal{D}' \sim \mathbf{D}$, the meta-learner can now generate an efficient predictor on task $\mathcal{D}'$.

Meta-learning with the sample compression hypernetwork. To turn the reconstruction hypernetwork $\mathcal{R}_\theta$ of Section 4 into a meta-learner, we propose to make the creation of the compression set and the message an explicit components of our learning algorithm via two functions: a *sample compressor* $\mathcal{C}_\phi$ and a *message compressor* $\mathcal{M}_\psi$, both taking a data matrix as an input. The sample compressor $\mathcal{C}_\phi$, parametrized by ϕ, outputs the product between a binary mask vector and the input data matrix, resulting in the compression set. The message compressor $\mathcal{M}_\psi$, parametrized by ψ, outputs a vector of the chosen message size b. See Figure 1 for a high-level depiction of the resulting architecture, and subsection 6.1 for implementation details.

Our goal is to learn parameters ϕ, ψ and θ such that, for any task $\mathcal{D}' \sim \mathbf{D}$ producing $S' \sim \mathcal{D}'$, the resulting output gives rise to a downstream predictor $h_{\gamma'}$ of low generalisation loss $\mathcal{L}_{\mathcal{D}'}(h_{\gamma'})$, with

$$\gamma' = \mathcal{R}_\theta\left(\mathcal{C}_\phi(S'), \mathcal{M}_\psi(S')\right). \tag{2}$$

Given a training meta-dataset $\mathbf{S} = \{S_i\}_{i=1}^n$, we propose to optimize the following objective:

$$\min_{\phi,\psi,\theta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i - |\hat{S}_i|} \sum_{(\mathbf{x},y)\in\hat{T}_i} \ell\big(h_{\gamma_i}(\mathbf{x}), y\big) \;\middle|\; \gamma_i = \mathcal{R}_\theta\left(\mathcal{C}_\phi(\hat{S}_i), \mathcal{M}_\psi(\hat{S}_i)\right) \right\}, \tag{3}$$

where, in conformity with classical meta-learning literature [29, 32], each task dataset S_i is split into a *support set* $\hat{S}_i \subset S_i$ and a *query set* $\hat{T}_i = S_i \setminus \hat{S}_i$; the former is used to learn the downstream network h_{γ_i} and the latter to compute h_{γ_i}'s loss. Note that this is a surrogate for Equation (1), as the complement of the compression set $S_{\bar{\mathbf{j}}}$ is replaced by the query set $\hat{T}_i$ in Equation (3). The corresponding learning algorithm is summarized by Algorithm 1.

Generalisation guarantees for encoder-decoder meta-learning. The meta-learner design described above is directly driven by the sample compression theory. Interestingly, it can be seen as an encoder-decoder model, with dual encoders $(\mathcal{C}_\phi, \mathcal{M}_\psi)$, and decoder $\mathcal{R}_\theta$, which comes with computable guarantees. Indeed, once the parameters (ϕ, ψ, θ) are learned from Equation (3), every downstream network $h_{\gamma'}$ obtained from Equation (2) allow a statistically valid upper bound on its generalisation loss $\mathcal{L}_{\mathcal{D}'}(h_{\gamma'})$, computable from either Theorems 1 and 2 (for discrete messages) or Theorem 3 (for continuous messages). Furthermore, the generalization bound can be computed on the union of the query set and the support set, excluding the compression set, since only the latter is given to the reconstruction function to generate the downstream network parameters.

6 Preliminary Experiments

Many architecture choices for the compressors and the reconstruction networks stem from the general design summarized by Figure 1. We describe our specific choices to conduct preliminary experiments in subsection 6.1. These are used to obtain the empirical results on a synthetic meta-learning problem presented in subsection 6.2.

Algorithm 1 Meta-Learning with the Sample Compression Hypernetwork

Inputs : $\mathbf{S} = \{S_i\}_{i=1}^n$, a meta-dataset
$\alpha \in \mathbb{N}$, support set size $(1 \leq \alpha < \min_i[m_i])$
$c, b \in \mathbb{N}$, the compression set and message size
BackProp, a function doing a gradient descent step
$\phi, \psi, \theta \leftarrow$ Initialize parameters
while Stopping criteria is not met **do**:
  for $i = 1, \ldots, n$ **do**:
    $\hat{S}_i \leftarrow$ Sample α datapoints from S_i
    $\mathbf{j} \leftarrow \mathcal{C}_\phi(\hat{S}_i)$ such that $|\,\mathbf{j}\,| = c$
    $\boldsymbol{\sigma} \leftarrow \mathcal{M}_\psi(\hat{S}_i)$ such that $|\boldsymbol{\sigma}| = b$
    $\gamma \leftarrow \mathcal{R}_\theta(\hat{S}_{i,\mathbf{j}}, \boldsymbol{\sigma})$
    loss $\leftarrow \frac{1}{m_i - \alpha} \sum_{(\mathbf{x},y) \in S_i \setminus \hat{S}_i} l\big(h_\gamma(\mathbf{x}), y\big)$
    $\phi, \psi, \theta \leftarrow$ BackProp(loss)
  end for
end while
return $\mathcal{C}_\phi, \mathcal{M}_\psi, \mathcal{R}_\theta$

6.1 Implementation details

In the following, we experiment in a simple binary classification setting, with features $\mathcal{X} = \mathbb{R}^d$ and labels $\mathcal{Y} = \{-1, 1\}$.

Message compressor network. We consider two versions of message compressor $\mathcal{M}_\psi$: the discrete and the continuous version, referring to the outputted message type. Both discrete and continuous message compressors require setting a message size b.

We first encode its input (a dataset) in a way that is permutation-independent way regarding the order of the examples in the dataset. Modules such as FSPool [33] or a transformer [31] ensure such property. Our experiments use a simpler mechanism described at the end of subsection 6.1 and referred to as Shared Transformation and Pool (STP). Then, a feedforward neural network is applied. In the discrete version, the final activation function of $\mathcal{M}_\psi$ is the *sign* function coupled with the straight-through estimator [15]. In the continuous version, the final activation function is the *tanh* function.

Sample compressor network. Given a fixed compression set size c, the sample compressor $\mathcal{C}_\phi$ is composed of c independent attention mechanisms. The queries are the result of an STP module, the keys are the result of a fully-connected network and the values are the feature values. Each attention mechanism outputs a probability distribution over the examples from the support set, and the example having the highest probability is added to the compression set.

Reconstruction hypernetwork. An STP module first handles the compression set outputted by $\mathcal{C}_\phi$, in order to encode it into a small vector and so that it is done in a permutation-independent way. Both the obtained compression set embedding and the message given by $\mathcal{M}_\psi$ are then fed to a feedforward neural network, whose output constitutes the parameters of the downstream network.

Shared Transformation and Pool (STP) module. This refers to a neural network component $\text{STP}_\omega(S_\mathbf{j})$ that maps a dataset, encoded by a data-matrix $\mathbf{X} \in \mathbb{R}^{|\mathbf{j}| \times d}$ and a binary label vector $\mathbf{y} \in \{-1, 1\}^{|\mathbf{j}|}$, into a fixed width embedding $\mathbf{z} \in \mathbb{R}^{d'}$. This embedding is obtained by first applying a fully-connected neural network $g_\omega : \mathbb{R}^d \to \mathbb{R}^{d'}$ to each row of $\mathbf{X}$, sharing the weights across rows, to obtain a matrix $\mathbf{M} \in \mathbb{R}^{|\mathbf{j}| \times d'}$ and then aggregating the result column-wise: $\mathbf{z} = \frac{1}{|\mathbf{j}|}\mathbf{M}^T\mathbf{y}$.

Bound computation. The generalization bound for discrete messages is computed from Theorem 1, using a uniform distribution over the messages of size b: $P_\Sigma(\boldsymbol{\sigma}) = 2^{-b}\ \forall \boldsymbol{\sigma} \in \{-1, 1\}^b$. For continuous messages, we rely on Theorem 3 with $\Delta(q, p) = \text{kl}(q, p) = q \cdot \ln \frac{q}{p} + (1 - q) \cdot \ln \frac{1-q}{1-p}$ (see Appendix A for details). In this case, we consider an isotropic b-dimensional Gaussian distribution of

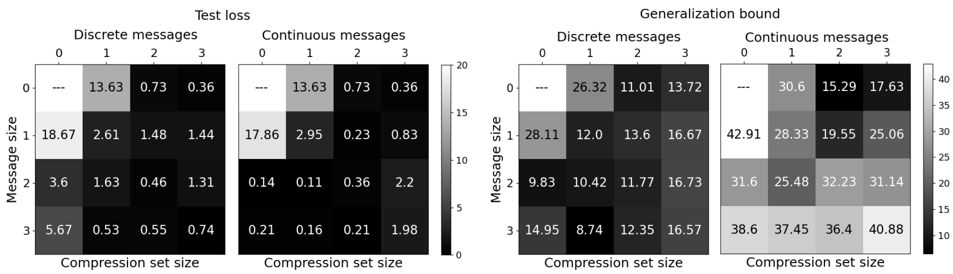

Figure 2: Average test risk (%) and generalization bound (%, with $\delta = 5\%$) on the *moons* meta-task.

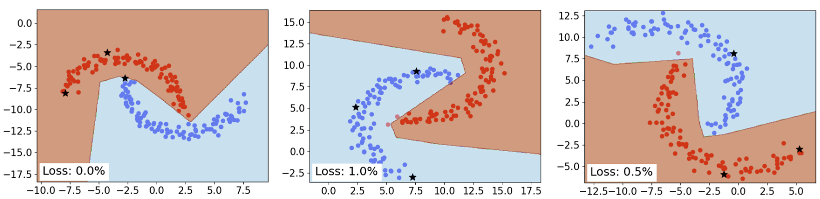

Figure 3: Examples of decision boundaries given by the downstream predictors, with a compression set of size 3 and without message, on test datasets. The stars show the retained points from the sample compressor $\mathcal{C}_\phi$. As shown by the axes, each plot is centered and scaled on the moons datapoints.

unit variance $P_\Sigma = \mathcal{N}(\mathbf{0}, \mathbf{I})$ as the prior distribution, and an isotropic Gaussian distribution centered on the message vector $Q_\Sigma = \mathcal{N}(\boldsymbol{\sigma}, \mathbf{I})$ as the posterior distribution, with $\boldsymbol{\sigma} \in [-1, 1]^b$.

Concerning the compression sets of fixed size c, given a dataset size m, we use $J = \{\mathbf{j} \in \mathcal{P}(\mathbf{m}) \,:\, |\,\mathbf{j}\,| = c\}$ and a uniform probability distribution over all distinct compression sets (sets that are not permutations of one another): $P_J(\mathbf{j}) = \binom{m}{c}^{-1} \,\forall \mathbf{j} \in J$.

6.2 Numerical results on a synthetic problem

We conduct our experiment on the *moons* 2-D synthetic dataset from Scikit-learn [27], which consists of two interleaving half circles with small Gaussian noise. We generate tasks by rotating (random degree in $[0, 360]$), translating (random moon center in $[-10, 10]^2$), and re-scaling the moons (random scaling factor in $[0.2, 5]$). We aim to determine how concise the compression set and the message must be to learn the task. We fixed the MLP architecture in the sample compressor, the message compressor, and the reconstruction function to a single-hidden layer MLP of size 100 while the predictor also is a single-hidden layer MLP of size 5. The *moons* meta-train set consists of 300 tasks of 200 examples, while the meta-test set consists of 100 tasks of 200 examples. We randomly split each dataset into support and query of equal size. See Appendix D for implementation details.

Figure 2 displays the average test zero-one loss and generalization bound for both discrete and continuous messages. The loss decreases as the compression set (c) and the message size (b) increase; interestingly, having these be too large simultaneously leads to worse performances than when a balance is found. A similar phenomenon occurs for the bound value, which finds its minimum for intermediate values of c and b. The continuous version of the algorithm leads to the best empirical results, whereas the discrete version leads to the best generalization bounds. It seems that the KL term's value in Theorem 3 is quite penalizing with respect to c: an interesting avenue is to regularize with regard to the KL term when training the meta-predictor. With both message types, we observe that tiny values for c and b are sufficient to encode variation in the *moons* datasets.

Figure 3 displays the decision boundaries on three different moon tasks of the predictors generated by our approach, with $c = 3$ and $b = 0$. We see that the sample compressor selects three examples far from each other, efficiently *compressing* the task, and which allows the hypernetwork *reconstructing* predictors of almost perfect accuracies.

7 Conclusion

We introduced a generalization bound for sample compression that permits the use of real-valued messages, and we developed a meta-learning algorithm that learns its reconstruction function to produce tight sample compression generalization bounds. Given the promising obtained results on a toy dataset, including tight generalization guarantees, we plan to pursue our experiments on real-life meta-learning tasks, including regression tasks.

Acknowledgements

This research is supported by the NSERC/Intact Financial Corporation Industrial Research Chair in Machine Learning for Insurance. Pascal Germain is supported by the Canada CIFAR AI Chair Program, and the NSERC Discovery grant RGPIN-2020-07223. Mathieu Bazinet is supported by a FRQNT B2X scholarship (343192).

Disclosure of Interests

The authors have no competing interests relative to the content of this article.

References

[1] Idan Attias, Steve Hanneke, Aryeh Kontorovich, and Menachem Sadigurschi. Agnostic sample compression schemes for regression. In *Forty-first International Conference on Machine Learning*, 2018.

[2] Jonathan Baxter. A model of inductive bias learning. *J. Artif. Intell. Res.*, 12:149–198, 2000.

[3] Mathieu Bazinet, Valentina Zantedeschi, and Pascal Germain. Sample compression unleashed : New generalization bounds for real valued losses, 2024.

[4] Marco C Campi and Simone Garatti. Compression, generalization and learning. *Journal of Machine Learning Research*, 24(339):1–74, 2023.

[5] Ofir David, Shay Moran, and Amir Yehudayoff. Supervised learning through the lens of compression. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[6] Sally Floyd and Manfred K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Mach. Learn.*, 21(3):269–304, 1995.

[7] Pascal Germain. Généralisations de la théorie pac-bayésienne pour l'apprentissage inductif, l'apprentissage transductif et l'adaptation de domaine. *PhD thesis*, 2015.

[8] Pascal Germain, Alexandre Lacasse, François Laviolette, Mario Marchand, and Jean-Francis Roy. Risk bounds for the majority vote: From a PAC-Bayesian analysis to a learning algorithm. *The Journal of Machine Learning Research*, 16:787–860, 2015.

[9] Pascal Germain, Alexandre Lacoste, François Laviolette, Mario Marchand, and Sara Shanian. A pac-bayes sample-compression approach to kernel methods. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 297–304. Omnipress, 2011.

[10] Steve Hanneke and Aryeh Kontorovich. Stable sample compression schemes: New applications and an optimal SVM margin bound. In *Algorithmic Learning Theory*, pages 697–721. PMLR, 2021.

[11] Steve Hanneke, Aryeh Kontorovich, and Menachem Sadigurschi. Efficient Conversion of Learners to Bounded Sample Compressors. *Proceedings of Machine Learning Research vol*, 75:1–21, 2018.

[12] Steve Hanneke, Aryeh Kontorovich, and Menachem Sadigurschi. Sample Compression for Real-Valued Learners. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pages 466–488. PMLR, 2019.

[13] Steve Hanneke, Shay Moran, and Waknine Tom. List sample compression and uniform convergence. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 2360–2388. PMLR, 2024.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034. IEEE Computer Society, 2015.

[15] Geoffrey Hinton. Neural networks for machine learning. coursera, video lectures, 2012.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[17] François Laviolette and Mario Marchand. PAC-Bayes risk bounds for sample-compressed Gibbs classifiers. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 481–488, Bonn, Germany, 2005. ACM Press.

[18] François Laviolette and Mario Marchand. PAC-Bayes Risk Bounds for Stochastic Averages and Majority Votes of Sample-Compressed Classifiers. *Journal of Machine Learning Research*, 8(7), 2007.

[19] François Laviolette, Mario Marchand, and Mohak Shah. Margin-sparsity trade-off for the set covering machine. In João Gama, Rui Camacho, Pavel Brazdil, Alípio Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005, 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005, Proceedings*, volume 3720 of *Lecture Notes in Computer Science*, pages 206–217. Springer, 2005.

[20] Nick Littlestone and Manfred K. Warmuth. Relating data compression and learnability. *Technical report*, University of California Santa Cruz, Santa Cruz, CA, 1986.

[21] Mario Marchand and John Shawe-Taylor. Learning with the set covering machine. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 345–352. Morgan Kaufmann, 2001.

[22] Mario Marchand and John Shawe-Taylor. The set covering machine. *Journal of Machine Learning Research*, 3(4-5):723–746, 2002.

[23] Mario Marchand and Marina Sokolova. Learning with decision lists of data-dependent features. *J. Mach. Learn. Res.*, 6:427–451, 2005.

[24] David A McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.

[25] Shay Moran and Amir Yehudayoff. Sample compression schemes for vc classes. *Journal of the ACM (JACM)*, 63(3):1–10, 2016.

[26] Dario Paccagnan, Marco Campi, and Simone Garatti. The pick-to-learn algorithm: Empowering compression for tight generalization bounds and improved post-training performance. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] 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.

[28] Benjamin I. P. Rubinstein and J. Hyam Rubinstein. A geometric approach to sample compression. *J. Mach. Learn. Res.*, 13:1221–1261, 2012.

[29] Amrith Setlur, Oscar Li, and Virginia Smith. Is support set diversity necessary for meta-learning? *CoRR*, abs/2011.14048, 2020.

[30] Mohak Shah. Sample compression bounds for decision trees. In Zoubin Ghahramani, editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 799–806. ACM, 2007.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[32] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3630–3638, 2016.

[33] Yan Zhang, Jonathon S. Hare, and Adam Prügel-Bennett. Fspool: Learning set representations with featurewise sort pooling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

A Corollaries of Theorem 2

For completeness, we present the corollaries of Theorem 2 derived by [3].

Corollary 1 ([3])**.** *In the setting of Theorem 2, with $\Delta(q,p) = \mathrm{kl}(q,p) = q \cdot \ln \frac{q}{p} + (1-q) \cdot \ln \frac{1-q}{1-p}$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^m$, we have:*

$$\forall \mathbf{j} \in J, \sigma \in M(|\mathbf{j}|) \;:\; \mathrm{kl}\Big(\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)), \mathcal{L}_D(\mathcal{R}(S_{\mathbf{j}},\sigma))\Big) \le \frac{1}{m-|\,\mathbf{j}\,|}\left[\ln\left(\frac{2\sqrt{m-|\,\mathbf{j}\,|}}{P_J(\mathbf{j}) \cdot P_{M(|\mathbf{j}|)}(\sigma)\cdot\delta}\right)\right].$$

The previous corollary is based on the use of Lemma 4.1 from Germain [7], leading to $\mathbb{E}_{T\sim\mathcal{D}^{m-|\mathbf{j}|}} e^{(m-|\,\mathbf{j}\,|)\cdot\Delta(\mathcal{L}_T(\mathcal{R}(S_{\mathbf{j}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma)))} \le 2\sqrt{m-|\,\mathbf{j}\,|}$.

Corollary 2 ([3])**.** *In the setting of Theorem 2, with $\Delta_C(q,p) = -\ln\big(1-(1-e^{-C})p\big) - Cq$ (where $C>0$), with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^m$, we have:*

$$\forall \mathbf{j} \in J, \sigma \in M(|\mathbf{j}|) \;:$$
$$\mathcal{L}_D(\mathcal{R}(S_{\mathbf{j}},\sigma)) \le \frac{1}{1-e^{-C}}\left[1 - \exp\left(-C\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)) + \frac{\ln\big(P_J(\mathbf{j})\cdot P_{M(|\mathbf{j}|)}(\sigma)\cdot\delta\big)}{m-|\mathbf{j}|}\right)\right]$$

Corollary 3 ([3])**.** *In the setting of Theorem 2, with $\Delta(q,p) = p - q$, for a loss function taking values in the interval $[a,b]$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^m$, we have:*

$$\forall \mathbf{j} \in J, \sigma \in M(|\mathbf{j}|) \;:\; \mathcal{L}_D(\mathcal{R}(S_{\mathbf{j}},\sigma)) \le \widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)) + \frac{(b-a)^2}{8} - \frac{\ln\big(P_J(\mathbf{j})\cdot P_{M(|\mathbf{j}|)}(\sigma)\cdot\delta\big)}{m-|\,\mathbf{j}\,|}$$

B Proof of Theorem 3

Theorem 3 (Sample compression - real-valued losses with continuous messages)**.** *For any distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, for any set $J \subseteq \mathcal{P}(\mathbf{m})$ such that $\max_{\mathbf{j}\in J}|\mathbf{j}| = c$, for any distribution P_J over J, for any prior distribution P_Σ over Σ, for any reconstruction function $\mathcal{R}$, for any loss $\ell : \mathcal{Y}\times\mathcal{Y} \to [0,1]$, for any convex function $\Delta : [0,1]\times[0,1] \to \mathbb{R}$ and for any $\delta \in (0,1]$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^m$, we have:*

$$\forall \mathbf{j} \in J, Q_\Sigma \textit{ over } \Sigma :$$
$$\Delta\left(\mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma} \widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)), \mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma} \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma))\right) \le \frac{1}{m-c}\left[\mathrm{KL}(Q_\Sigma||P_\Sigma) + \ln\left(\frac{\mathcal{J}_\Delta(m-c)}{P_J(\mathbf{j})\cdot\delta}\right)\right],$$

with

$$\mathcal{J}_\Delta(m-c) = \mathop{\mathbb{E}}_{\sigma\sim P_\Sigma}\; \mathop{\mathbb{E}}_{T_{\mathbf{j}}\sim\mathcal{D}^{|\mathbf{j}|}}\; \mathop{\mathbb{E}}_{T_{\bar{\mathbf{j}}}\sim\mathcal{D}^{m-|\mathbf{j}|}} e^{(m-c)\cdot\Delta\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{j}}}}(\mathcal{R}(T_{\mathbf{j}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{j}},\sigma))\right)}.$$

Proof. Let J be a set of indices, such that $\max_{\mathbf{j}\in J}|\mathbf{j}| = c$. Let $\mathbf{j} \in J$ be a given index set. Let $\mathcal{Q}_\Sigma$ be the space of probability distribution over Σ. Our first goal is to bound the distance between the true risk and the empirical risk $\eta\Delta\Big(\mathbb{E}_{\sigma\sim Q_\Sigma}\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)), \mathbb{E}_{\sigma\sim Q_\Sigma}\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma))\Big)$ with $\eta = m - \max_{\mathbf{j}\in J}|\mathbf{j}| = m-c$.

$$\forall Q_\Sigma \text{ over } \Sigma :$$
$$(m-c)\Delta\left(\mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)), \mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma))\right)$$
$$\le \mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}(m-c)\Delta\Big(\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)), \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma))\Big) \qquad \text{(Jensen's Inequality)}$$
$$\le \mathrm{KL}(Q_\Sigma||P_\Sigma) + \ln\left(\mathop{\mathbb{E}}_{\sigma\sim P_\Sigma} e^{(m-c)\Delta\left(\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma))\right)}\right)$$
$$\text{(Change of measure)}$$

Using Markov's Inequality, we know that with probability at least $1-\delta_{\mathbf{j}}$, where $\delta_{\mathbf{j}} \in (0,1)$, over the sampling of $S \sim \mathcal{D}^m$, we have for all Q_Σ over Σ:

$$
\begin{aligned}
&(m-c)\Delta\left(\mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma} \widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)), \mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma} \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma))\right) \\
&\leq \mathrm{KL}(Q_\Sigma||P_\Sigma) + \ln\left(\frac{1}{\delta}\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathop{\mathbb{E}}_{\sigma\sim P_\Sigma} e^{(m-c)\Delta\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{j}}}}(\mathcal{R}(T_{\mathbf{j}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{j}},\sigma))\right)}\right)
\end{aligned}
$$

We now wish to invert the expectations in the rightmost term to obtain: $\mathbb{E}_{T\sim\mathcal{D}^m}\mathbb{E}_{\sigma\sim P_\Sigma} = \mathbb{E}_{\sigma\sim P_\Sigma}\mathbb{E}_{T\sim\mathcal{D}^m}$. In most cases, for a data-dependent predictor, this equality does not hold. However, we defined the prior P_Σ over Σ.

We use the independence of the prior to T and the *i.i.d.* assumption to separate $T_{\mathbf{j}}$ and $T_{\bar{\mathbf{j}}} = T \setminus T_{\mathbf{j}}$:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathop{\mathbb{E}}_{\sigma\sim P_\Sigma} e^{(m-c)\Delta\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{j}}}}(\mathcal{R}(T_{\mathbf{j}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{j}},\sigma))\right)} \\
&= \mathop{\mathbb{E}}_{\sigma\sim P_\Sigma}\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m} e^{(m-c)\Delta\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{j}}}}(\mathcal{R}(T_{\mathbf{j}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{j}},\sigma))\right)} \qquad \text{(Independence of the prior from } S\text{)} \\
&= \mathop{\mathbb{E}}_{\sigma\sim P_\Sigma}\mathop{\mathbb{E}}_{T_{\mathbf{j}}\sim\mathcal{D}^{|\mathbf{j}|}}\mathop{\mathbb{E}}_{T_{\bar{\mathbf{j}}}\sim\mathcal{D}^{m-|\mathbf{j}|}} e^{(m-c)\Delta\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{j}}}}(\mathcal{R}(T_{\mathbf{j}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{j}},\sigma))\right)} \\
&= \mathcal{J}_\Delta(m-c).
\end{aligned}
$$

Let $\delta = \sum_{\mathbf{j}\in J^{(c)}} \delta_{\mathbf{j}}$, with $\delta_{\mathbf{j}} = P_{J^{(c)}}(\mathbf{j})\cdot\delta$. By the union bound, we obtain the desired result. □

C Corollaries of Theorem 3

The following corollaries are easily derived by choosing a comparator function Δ and bounding $\mathcal{J}_\Delta(m-c)$.

Corollary 4. *In the setting of Theorem 3, with $\Delta(q,p) = \mathrm{kl}(q,p) = q\cdot\ln\frac{q}{p} + (1-q)\cdot\ln\frac{1-q}{1-p}$, with probability at least $1-\delta$ over the draw of $S\sim\mathcal{D}^m$, for all $\mathbf{j}\in J$ and posterior probability distribution Q_Σ, we have:*

$$
\mathrm{kl}\left(\mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)), \mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma))\right) \leq \frac{1}{m-c}\left[\mathrm{KL}(Q_\Sigma||P_\Sigma) + \ln\left(\frac{2\sqrt{m-c}}{P_J(\mathbf{j})\cdot\delta}\right)\right]
$$

Corollary 5. *In the setting of Theorem 3, with $\Delta_C(q,p) = -\ln\big(1-(1-e^{-C})p\big) - Cq$ (where $C>0$), with probability at least $1-\delta$ over the draw of $S\sim\mathcal{D}^m$, for all $\mathbf{j}\in J$ and posterior probability distribution Q_Σ, we have:*

$$
\mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma)) \leq \frac{1}{1-e^{-C}}\left[1-\exp\left(-C\mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)) - \frac{\mathrm{KL}(Q_\Sigma||P_\Sigma) - \ln\left(P_J(\mathbf{j})\cdot\delta\right)}{m-c}\right)\right]
$$

Corollary 6. *In the setting of Theorem 3, with $\Delta(q,p) = \lambda(p-q)$, for a loss function taking values in the interval $[a,b]$, with probability at least $1-\delta$ over the draw of $S\sim\mathcal{D}^m$, for all $\mathbf{j}\in J$ and posterior probability distribution Q_Σ, we have:*

$$
\mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{j}},\sigma)) \leq \mathop{\mathbb{E}}_{\sigma\sim Q_\Sigma}\widehat{\mathcal{L}}_{S_{\bar{\mathbf{j}}}}(\mathcal{R}(S_{\mathbf{j}},\sigma)) + \frac{\lambda(b-a)^2}{8(m-c)} + \frac{\mathrm{KL}(Q_\Sigma||P_\Sigma) - \ln\left(P_J(\mathbf{j})\cdot\delta\right)}{\lambda}
$$

D Numerical experiment and implementation details

We fixed the batch size to 20. We added skip connections and batch norm in both the modules of the meta-learner and the predictor to accelerate the training time. The experiments were conducted using an NVIDIA GeForce RTX 2080 Ti graphic card.

We used the Adam optimizer [16] and trained for at most 200 epochs, stopping when the validation accuracy did not diminish for 20 epochs. We initialized the weights of each module using the Kaiming uniform technique [14].