Sample Compression Unleashed: New Generalization Bounds for Real Valued Losses

Mathieu Bazinet
Université Laval

Valentina Zantedeschi
ServiceNow Research, Université Laval

Pascal Germain
Université Laval

Abstract

The sample compression theory provides generalization guarantees for predictors that can be fully defined using a subset of the training dataset and a (short) message string, generally defined as a binary sequence. Previous works provided generalization bounds for the zero-one loss, which is restrictive notably when applied to deep learning approaches. In this paper, we present a general framework for deriving new sample compression bounds that hold for real-valued unbounded losses. Using the Pick-To-Learn (P2L) meta-algorithm, which transforms the training method of any machine-learning predictor to yield sample-compressed predictors, we empirically demonstrate the tightness of the bounds and their versatility by evaluating them on random forests and multiple types of neural networks.

1 INTRODUCTION

Sample compression theory, introduced by Littlestone and Warmuth (1986), is based on the fundamental idea that "compressing implies learning" (David et al., 2016). If it is possible to provably show that a learned model can be completely defined by a subset of the training dataset, then sample compression theory gives us generalization guarantees. The most well-known learning algorithms that comply with the sample compression framework are the support vector machine (SVM) (Boser et al., 1992) and the perceptron (Rosenblatt, 1958; Moran et al., 2020); the relevant training subset being formed by the support vectors in the former case, and the points causing an update of the predictor in the latter case. More recently, Snyder and Vishwanath (2020) and Paccagnan et al. (2024) have introduced the first sample compression results for neural networks.

The sample compression theory is rich and multiple different approaches exist. For example, Attias et al. (2018, 2023, 2024); Ben-David et al. (2024); David et al. (2016); Floyd and Warmuth (1995); Hanneke and Kontorovich (2021); Hanneke et al. (2018, 2019, 2024); Moran and Yehudayoff (2016); Rubinstein and Rubinstein (2012) propose theoretical results relating the VC dimension (Vapnik and Chervonenkis, 1971) and the compression analysis. By relating the probability of *change of compression* to the true risk, Campi and Garatti (2023); Paccagnan et al. (2024) express very tight guarantees for the consistent case, i.e., when the error on the training set is zero. Finally, Marchand and Shawe-Taylor (2002); Marchand et al. (2003); Graepel et al. (2005); Laviolette et al. (2005, 2009); Marchand and Sokolova (2005); Hussain et al. (2007); Shah (2007) give computable risk certificates valid even in the non-consistent case.

In this paper, we build on the setting of Laviolette et al. (2005). Their sample-compression bound is based on the binomial test-set bound of Langford (2005), which by definition is the tightest test-set bound for the zero-one loss under the sole *i.i.d.* assumption. However, the use of the zero-one loss restricts its application to supervised classification problems. By leveraging proof techniques from the PAC-Bayesian literature, we extend the framework to real-valued losses and open the way to obtaining bounds directly for the cross-entropy loss (Pérez-Ortiz et al., 2021) and unbounded losses (Haddouche et al., 2021; Casado Telletxea et al., 2025; Rodríguez-Gálvez et al., 2024), for example under the sub-Gaussian assumption (Kahane, 1960). Finally, we train deep neural networks and random forests with Pick-To-Learn (P2L) (Paccagnan et al., 2024), a meta-algorithm that modifies the training loop of a model to yield a sample-compressed predictor, and assess the tightness of our bounds in different settings.

Of note, a major asset of our sample-compress bounds is that they do not depend on the number of learnable parameters. Two models of different sizes can achieve the same guarantees as long as they achieve the same

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

empirical loss using the same amount of data. This lets us train large models such as DistilBERT (Sanh et al., 2019) and still achieve tight generalization bounds.

The paper is organized as follows. In Section 2, we present the sample compression theory and the meta-algorithm Pick-To-Learn (P2L) (Paccagnan et al., 2024). In Section 3, we first present Theorem 3, a new general sample-compression theorem that holds for any real-valued losses. Leveraging the comparator functions of PAC-Bayes theory (McAllester, 1998), we present two new sample compression bounds for losses in the interval unit, Corollary 4 and Corollary 6, which respectively yield the tightest bound in theory and in practice. Then, we present Corollary 7, which holds for any unbounded losses, under the assumption that the moment-generating function is bounded. We finish this section by proving the tightness of our results over the previous state-of-the-art sample compression bound. Finally, in Section 4, we empirically show the tightness of our bounds by training deep neural networks on image and text classification problems with P2L. We adapt P2L to regression, train regression trees and forests with this modified algorithm and provide the first sample compression generalization bounds for tree-based regression predictors.

2 BACKGROUND AND NOTATION

We are interested in the supervised learning framework. Let $(\boldsymbol{x}_1,y_1),...,(\boldsymbol{x}_n,y_n)$ be a sequence of n datapoints sampled *i.i.d.* (independently and identically distributed) from an unknown distribution $\mathcal{D}$ over $\mathbb{R}^d\times\mathcal{Y}$. The dataset $S=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^n$ is generated with the sequence of datapoints.[1] The targets are defined by the task at hand, with $\mathcal{Y}\in\{-1,+1\}$ for binary classification tasks and $\mathcal{Y}\subseteq\mathbb{R}$ for regression tasks. In this section, we focus on binary classification problems, but in Section 3, we study both classification and regression settings.

Let $\mathcal{H}$ be a family of predictors $h:\mathcal{X}\to\mathcal{Y}$. Let $A:\bigcup_{k=1}^{\infty}(\mathcal{X}\times\mathcal{Y})^k\to\mathcal{H}$ be a learning algorithm that takes a dataset S and returns a predictor $A(S)$. We consider the zero-one loss function $\ell^{0\text{-}1}(h,\boldsymbol{x},y)=\mathbb{I}[h(\boldsymbol{x})\neq y]$, with $\mathbb{I}[a]=1$ if the predicate a is true and 0 otherwise. Then, the true risk of the hypothesis h is defined as

$$R_{\mathcal{D}}(h)=\mathop{\mathbb{P}}_{(\boldsymbol{x},y)\sim\mathcal{D}}(h(\boldsymbol{x})\neq y)=\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\mathbb{I}[h(\boldsymbol{x})\neq y]$$

and, for a realization $S\sim\mathcal{D}^n$, its empirical risk is defined as $\widehat{R}_S(h)=\frac{1}{n}\sum_{i=1}^n\mathbb{I}[h(\boldsymbol{x}_i)\neq y_i]$.

Since the distribution $\mathcal{D}$ is unknown, the true risk of a hypothesis cannot be computed. However, it can be upper bounded with high probability, using generalization bounds derived from statistical learning theories such as the sample compression theory.

2.1 Sample compression theory

Let the predictor $h=A(S)$ be the output of a learning algorithm A applied to a dataset S. In order to obtain guarantees on the generalization performance of h using the sample compression theory, we need to be able to uniquely define h as a function (the reconstruction function) of a subset of S (the compression set) and a complementary sequence of information (the message).

The compression set $S_{\mathbf{i}}$ is defined using a vector of indices $\mathbf{i}=\big(i_1,i_2,...,i_{|\mathbf{i}|}\big)$, where the indices are ordered such that $1\leq i_1<i_2<...<i_{|\mathbf{i}|}\leq n$. The vector $\mathbf{i}$ belongs in the set of all possible vectors composed of the natural numbers 1 through n, denoted

$$\mathcal{P}(n)=\Big\{\emptyset,\{1\},\{2\},...,\{n\},\{1,2\},...,\{1,n\},...,\{1,2,...,n\}\Big\}.$$

Using this notation, $\mathbf{i}$ indicates the datapoints of S that are present in $S_{\mathbf{i}}$:

$$S_{\mathbf{i}}=\big\{(\boldsymbol{x}_{i_1},y_{i_1}),...,(\boldsymbol{x}_{i_{|\mathbf{i}|}},y_{i_{|\mathbf{i}|}})\big\}\subseteq S.$$

Moreover, we define the complement vector $\mathbf{i}^c\in\mathcal{P}(n)$ such that $S_{\mathbf{i}^c}=S\setminus S_{\mathbf{i}}$ and $|\mathbf{i}^c|=n-|\mathbf{i}|$.

The message σ is chosen in a set $M(\mathbf{i})$, which contains all relevant messages associated to the compression set $\mathbf{i}$. The message is a complementary source of information needed to reconstruct the predictor.

A predictor h is called a sample-compressed predictor if there exists a vector $\mathbf{i}\in\mathcal{P}(n)$ and (optionally) a message $\sigma\in M(\mathbf{i})$ such that $h=\mathcal{R}(S_{\mathbf{i}},\sigma)$, where $\mathcal{R}:\bigcup_{m\leq n}(\mathcal{X}\times\mathcal{Y})^m\times\bigcup_{\mathbf{i}\in\mathcal{P}(n)}M(\mathbf{i})\to\overline{\mathcal{H}}$ is a data-independent deterministic reconstruction function and $\overline{\mathcal{H}}\subseteq\mathcal{H}$ is a discrete set of sample-compressed predictors.

In this paper, we distinguish two categories of reconstruction functions: inherent and dedicated. An inherent reconstruction function is used when a learning algorithm A is its own reconstruction function. An algorithm A is an inherent reconstruction function when, given a dataset S and its compression set $S_{\mathbf{i}}$, the following equality holds $A(S)=A(S_{\mathbf{i}})$. The most well-known example of an inherent reconstruction function is the SVM (Boser et al., 1992). Other examples of inherent reconstruction functions are the perceptron (Rosenblatt, 1958) and Pick-To-Learn (Paccagnan et al., 2024). On the other hand, dedicated reconstruction functions are used when an algorithm cannot be used to reconstruct the learned predictor from a compression set. The dedicated reconstruction function is different

[1] In this paper, we do not consider repeated datapoints. However, all definitions and results could be easily adapted to use multisets to account for repetitions, similarly to the work of Campi and Garatti (2023).

from the learning algorithm A and is generally handcrafted to suit A. The reconstruction function of the SCM (Marchand and Shawe-Taylor, 2002) and all its iterations (e.g., Marchand et al., 2003; Marchand and Sokolova, 2005; Laviolette et al., 2005; Kestler et al., 2006; Hussain et al., 2007; Drouin et al., 2019) are examples of dedicated reconstruction functions.

We provide an example of a dedicated reconstruction function for a very simple predictor, the decision stump.

Example (Shah et al. (2011)). *Given a datapoint $\boldsymbol{x}' = (x'_1,...,x'_d)$, a direction $\diamond \in \{-1,+1\}$ and an index $1 \leq k \leq d$, the stump is defined $f_{(\boldsymbol{x}',d,k)}(\boldsymbol{x}) = \mathbb{I}[\diamond \cdot (x_k - x'_k) > 0]$. To learn a decision stump over a dataset S, each combination of $\boldsymbol{x}' \in S$, $\diamond \in \{-1,+1\}$ and $0 \leq k \leq d$ is tested. Once the decision stump is learned, it is completely defined by the datapoint $\boldsymbol{x}'$, the direction $\diamond$ and the index k. Our compression set is $S_{\mathbf{i}} = \{\boldsymbol{x}'\}$ and the message is $\sigma = \{\diamond, k\}$. With the compression set and the message, we can fully reconstruct the stump with $\mathcal{R}(S_{\mathbf{i}}, \{\diamond, k\}) = f_{(S_{\mathbf{i}},\diamond,k)}$.*

Let $P_{\overline{\mathcal{H}}}$ be a distribution over $\overline{\mathcal{H}}$, such that $\sum_{h \in \overline{\mathcal{H}}} P_{\overline{\mathcal{H}}}(h) \leq 1$. As all sample-compressed predictors are uniquely defined using the index vector and the message, we choose the distribution $P_{\overline{\mathcal{H}}}$ to be a product of two distributions $P_{\overline{\mathcal{H}}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) = P_{\mathcal{P}(n)}(\mathbf{i}) P_{M(\mathbf{i})}(\sigma)$, with $P_{\mathcal{P}(n)}$ a distribution on $\mathcal{P}(n)$ and $P_{M(\mathbf{i})}$ a distribution on $M(\mathbf{i})$. Following previous works (e.g. Marchand and Sokolova, 2005), we require the distribution $P_{\mathcal{H}}$ to be data-independent, in order to avoid further assumptions. Without any information on the data, we generally set $P_{M(\mathbf{i})}$ to a uniform distribution. As for the distribution $P_{\mathcal{P}(n)}$, it is usually set to penalize larger compression sets (Laviolette et al., 2005; Marchand and Shawe-Taylor, 2002; Marchand and Sokolova, 2005). For any size of compression set $|\mathbf{i}|$, there are $\binom{n}{|\mathbf{i}|}$ different possible compression sets. We set the distribution $P_{\mathcal{P}(n)}(\mathbf{i})$ to be $\binom{n}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|)$, with $\zeta(m) = \frac{6}{\pi^2}(m+1)^{-2}$. This choice is discussed by Marchand and Sokolova (2005).

We now present the sample compression bound of Laviolette et al. (2005). This result is derived using the binomial test-set bound of Langford (2005), which by definition is the tightest test-set bound for the zero-one loss under the sole i.i.d. assumption.

Theorem 1 (Laviolette et al. (2005), Theorem 1). *For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, for any family of set of messages $\{M(\mathbf{i}) | \mathbf{i} \in \mathcal{P}(n)\}$, for any deterministic reconstruction function $\mathcal{R}$ that outputs sample-compressed predictors $h \in \overline{\mathcal{H}}$ and for any $\delta \in (0,1]$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^n$, we have*

$$\forall \mathbf{i} \in \mathcal{P}(n), \sigma \in M(\mathbf{i}) : \quad R_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) \leq \overline{\mathrm{Bin}}\left(\kappa, |\mathbf{i}^c|, \binom{n}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|) P_{M(\mathbf{i})}(\sigma)\delta\right),$$

with $\kappa = |\mathbf{i}^c| \widehat{R}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma))$ and

$$\overline{\mathrm{Bin}}(k,m,\delta) = \sup_{r \in [0,1]} \left\{ \sum_{i=0}^{k} \binom{m}{i} r^i (1-r)^{m-i} \geq \delta \right\}.$$

This theorem can be applied to any family of sample-compressed predictors, such as the support vector machine (Boser et al., 1992), the perceptron (Rosenblatt, 1958) and the set covering machine (Marchand and Shawe-Taylor, 2002). To apply this theorem to neural networks, one must design a reconstruction function outputting neural networks. To this end, Snyder and Vishwanath (2020) propose to reparameterize a 2-layer LeakyReLU network in order to obtain "support vectors", which become the compression set of the reconstructed network. The following section presents a more general approach proposed by Paccagnan et al. (2024).

2.2 Pick-To-Learn

Conceptualized by Paccagnan et al. (2024), Pick-To-Learn (P2L) is a model-agnostic meta-algorithm that trains any model in such a way that it becomes a sample-compressed predictor. This algorithm is specifically designed for the generalization bound of Campi and Garatti (2023), which holds only for sample compressed predictors in the *consistent case*, i.e., when $\widehat{R}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) = 0$.

To obtain sample-compressed predictors, P2L iteratively builds the compression set and trains the model on it. Starting with an initial predictor h_0, P2L tests the model on the whole dataset, picks the datapoint over which the model got the largest loss value, and adds it to the compression set. Then, using a learning algorithm A, P2L trains the model on the newly created compression set. The previous steps are repeated until the model achieves zero errors on the training set $S_{\mathbf{i}^c}$ (excluding the compression set datapoints), which is equivalent to stopping when the cross-entropy loss ($\ell^{\text{x-e}}$) becomes smaller than $-\ln(0.5)$. We present P2L in Algorithm 1.

Algorithm 1: Pick-To-Learn (P2L)

Initialize : $S_{\mathbf{i}} \leftarrow \emptyset$
Initialize : $h_{\mathbf{i}} \leftarrow h_0$
Initialize : $(\overline{\boldsymbol{x}}, \overline{y}) \leftarrow \mathrm{argmax}_{(\boldsymbol{x},y) \in S} \ell^{\text{x-e}}(h_0, \boldsymbol{x}, y)$
while $-\ln(0.5) \leq \ell^{\text{x-e}}(h_{\mathbf{i}}, \overline{\boldsymbol{x}}, \overline{y})$ **do**
  $S_{\mathbf{i}} \leftarrow S_{\mathbf{i}} \cup \{(\overline{\boldsymbol{x}}, \overline{y})\}$
  $h_{\mathbf{i}} \leftarrow A(S_{\mathbf{i}})$
  $(\overline{\boldsymbol{x}}, \overline{y}) \leftarrow \mathrm{argmax}_{(\boldsymbol{x},y) \in S_{\mathbf{i}^c}} \ell^{\text{x-e}}(h_{\mathbf{i}}, \boldsymbol{x}, y)$
end
return $h_{\mathbf{i}}$

Leveraging from the theoretical results of Campi and Garatti (2023), Paccagnan et al. (2024) derived a theorem specifically for the P2L algorithm.

Theorem 2 (Paccagnan et al. (2024), Theorem 4.2)**.** *Let $h_{\mathbf{i}} = \mathcal{R}(S_{\mathbf{i}}, \emptyset)$ be the output of P2L. For any $\delta \in (0,1]$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^n$, we have*

$$R_{\mathcal{D}}(h_{\mathbf{i}}) \leq \bar{\varepsilon}(|\mathbf{i}|, \delta),$$

where, for $k = 0,1,\ldots,n-1$, $\bar{\varepsilon}(k,\delta)$ is the unique solution to the equation $\Psi_{k,\delta}(\varepsilon) = 1$ in the interval $[\frac{k}{n}, 1]$, with

$$\Psi_{k,\delta}(\varepsilon) = \frac{\delta}{2n} \sum_{m=k}^{n-1} \frac{\binom{m}{k}}{\binom{n}{k}} (1-\varepsilon)^{-(n-m)} + \frac{\delta}{6n} \sum_{m=n+1}^{4n} \frac{\binom{m}{k}}{\binom{n}{k}} (1-\varepsilon)^{-(n-m)},$$

and $\bar{\varepsilon}(n,\delta) = 1$.

Note that the value of the previous bound is completely determined by $|\mathbf{i}|$, the size of the compression set. The faster P2L obtains zero errors (in terms of the number of iterations performed by Algorithm 1), the better the bound.

3 A GENERAL SAMPLE-COMPRESSION BOUND

Let $\mathcal{H}$ be a family of predictors $h : \mathcal{X} \to \overline{\mathcal{Y}}$, where $\overline{\mathcal{Y}} \supseteq \mathcal{Y}$ is a convex hull of $\mathcal{Y}$. For example, $[-1, 1]$ is the convex hull of $\{-1, +1\}$. We consider a loss function $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Then, the true risk of the hypothesis h is defined as $\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell(h, \boldsymbol{x}, y)$ and, for a realization $S \sim \mathcal{D}^n$, its empirical risk is defined as $\widehat{\mathcal{L}}_S(h) = \frac{1}{n}\sum_{i=1}^n \ell(h, \boldsymbol{x}_i, y_i)$. This setting is a generalization of the setting of Section 2. As Theorem 1 only holds for the zero-one loss, we need new results to extend the sample-compression theory to this setting.

To extend the work of Laviolette et al. (2005) to real-valued losses, we introduce a *comparator function* $\Delta : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and provide a new result inspired by the general PAC-Bayes bound (Germain et al., 2009). Theorem 3 presents a new general sample-compress bound that holds for any real-valued losses, extending the applicability of the sample-compression theory. The theorem is followed by a proof sketch highlighting the main steps, and the full proof is given in Appendix C.

Theorem 3. *For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, for any family of set of messages $\{M(\mathbf{i}) \mid \mathbf{i} \in \mathcal{P}(n)\}$, for any deterministic reconstruction function $\mathcal{R}$ that outputs sample-compressed predictors $h \in \overline{\mathcal{H}}$, for any loss $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, for any comparator 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 $S \sim \mathcal{D}^n$, we have*

$$\forall \mathbf{i} \in \mathcal{P}(n), \sigma \in M(\mathbf{i}): \Delta\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)), \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}}, \sigma))\Big) \leq \frac{1}{|\mathbf{i}^c|}\left[\log\binom{n}{|\mathbf{i}|} + \log\left(\frac{\mathcal{E}_\Delta(\mathbf{i}, \sigma)}{\zeta(|\mathbf{i}|) P_{M(\mathbf{i})}(\sigma)\delta}\right)\right],$$

with

$$\mathcal{E}_\Delta(\mathbf{i}, \sigma) = \mathop{\mathbb{E}}_{T_{\mathbf{i}} \sim \mathcal{D}^{|\mathbf{i}|}} \mathop{\mathbb{E}}_{T_{\mathbf{i}^c} \sim \mathcal{D}^{|\mathbf{i}^c|}} e^{|\mathbf{i}^c| \Delta\left(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}}, \sigma)), \mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}}, \sigma))\right)}.$$

Proof Sketch. For all $\mathbf{i} \in \mathcal{P}(n)$, $\sigma \in M(\mathbf{i})$, $\epsilon > 0$, using Chernoff's bound with $t > 0$, we have

$$\begin{aligned} &\mathop{\mathbb{P}}_{S\sim\mathcal{D}^n}\Big(\Delta\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)), \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}}, \sigma))\Big) > \epsilon\Big) \qquad (1)\\ &\leq e^{-t\epsilon} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n} e^{t\Delta\left(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)), \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}}, \sigma))\right)}\\ &= e^{-t\epsilon} \mathop{\mathbb{E}}_{S_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}} \mathop{\mathbb{E}}_{S_{\mathbf{i}^c}\sim\mathcal{D}^{|\mathbf{i}^c|}} e^{t\Delta\left(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)), \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}}, \sigma))\right)} \end{aligned}$$

where the last equality requires *i.i.d.* datapoints. For any $\delta_{\mathbf{i}}^\sigma \in (0,1]$, we define

$$\delta_{\mathbf{i}}^\sigma = e^{-t\epsilon} \mathop{\mathbb{E}}_{S_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}} \mathop{\mathbb{E}}_{S_{\mathbf{i}^c}\sim\mathcal{D}^{|\mathbf{i}^c|}} e^{t\Delta\left(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)), \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}}, \sigma))\right)} \qquad (2)$$

and solve for ϵ, using $t = |\mathbf{i}^c|$. The obtained solution is used to replace the ϵ in Eq. (1), which gives a bound valid with probability $\delta_{\mathbf{i}}^\sigma$ for every single predictor $\mathcal{R}(S_{\mathbf{i}}, \sigma)$. By setting $\delta_{\mathbf{i}}^\sigma = P_{\mathcal{P}(n)}(\mathbf{i}) P_{M(\mathbf{i})}(\sigma)\delta$ and applying a union bound over all $\mathbf{i} \in \mathcal{P}(n)$, $\sigma \in M(\mathbf{i})$, the final result holds uniformly with probability δ for all predictors outputted by $\mathcal{R}$. □

Theorem 3 holds for any comparator function Δ such that $\mathcal{E}_\Delta$ is finite for any pair $(\mathbf{i}, \sigma)$. Although bounding $\mathcal{E}_\Delta$ can be challenging, it was extensively studied for convex functions in PAC-Bayesian theory (e.g., McAllester, 1998; Maurer, 2004; Casado Telletxea et al., 2025; Hellström and Guedj, 2024). We leverage this theory and present novel corollaries for the three most well-known comparators.

First of all, we present a bound using the comparator $\Delta_C(q, p) = -\ln\big(1 - p(1 - e^{-C})\big) - Cq$. The family of bounds $\{\Delta_C : C > 0\}$ is commonly referred to as "Catoni bounds" (Catoni, 2007) in the PAC-Bayes literature.

Corollary 4. *In the setting of Theorem 3, for any $C > 0$, for any loss function $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to [0,1]$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^n$, we have*

$$\forall \mathbf{i} \in \mathcal{P}(n), \sigma \in M(\mathbf{i}) : \mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)) \leq \frac{1 - \exp(-\epsilon_C(\mathbf{i}, \sigma, \delta))}{1 - e^{-C}},$$

with

$$\epsilon_C(\mathbf{i}, \sigma, \delta) = C\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)) + \frac{1}{n - |\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|} + \log\left(\frac{1}{\zeta(|\mathbf{i}|) P_{M(\mathbf{i})}(\sigma)\delta}\right)\right].$$

For $0 \leq q,p \leq 1$, there exists $C^* = \operatorname{argsup}_{C>0} \Delta_C(q,p)$ such that Δ_{C^*} gives the tightest PAC-Bayesian bounds (Foong et al., 2021). This result also holds true for Theorem 3, when restricted to proper, convex and lower semicontinuous comparator functions $\Delta : [0,1]\times[0,1] \to \mathbb{R}$. Unfortunately, the Δ_C bound hold for only one value of C, chosen prior to seeing S. With a union bound argument, we can consider multiple parameters C simultaneously, but there is no guarantee that C^* is in this set. To circumvent this problem, we can use the binary Kullback-Leibler divergence comparator function $\mathrm{kl}(q,p) = q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}$, which is equivalent to $\Delta_{C^*}(q,p)$, as per the following proposition.

Proposition 5 (Germain et al. (2009), Proposition 2.1)**.** *For any $0 \leq q \leq p < 1$, we have $\sup_{C\geq 0} \Delta_C(q,p) = \mathrm{kl}(q,p)$.*

In practice, even with the term $1 = \mathcal{E}_{\Delta_C}(\mathbf{i},\sigma) \leq \mathcal{E}_{\mathrm{kl}}(\mathbf{i},\sigma) = 2\sqrt{n-|\mathbf{i}|}$, the kl bound stated below (Corollary 6) usually yield tighter bounds than the Δ_C bound (Corollary 4), as the optimal value C^* is unlikely to be selected before computing the bound. Moreover, the kl is known to be optimal for [0,1]-valued losses, as per the results of Hellström and Guedj (2024).

Corollary 6. *In the setting of Theorem 3, for any loss function $\ell : \mathcal{H}\times\mathcal{X}\times\mathcal{Y} \to [0,1]$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^n$, we have*

$$\forall \mathbf{i} \in \mathcal{P}(n), \sigma \in M(\mathbf{i}):$$
$$\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) \leq \mathrm{kl}^{-1}\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)), \epsilon_{\mathrm{kl}}(\mathbf{i},\sigma,\delta)\Big),$$

with $\mathrm{kl}^{-1}(q,\epsilon) = \operatorname{argsup}_{0\leq p\leq 1}\{\mathrm{kl}(q,p) \leq \epsilon\}$ *and*

$$\epsilon_{\mathrm{kl}}(\mathbf{i},\sigma,\delta) = \frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|} + \log\left(\frac{2\sqrt{n-|\mathbf{i}|}}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right].$$

Both Corollary 4 and Corollary 6 hold for losses bounded in [0,1]. Using the linear function $\Delta_\lambda(q,p) = \lambda(p-q)$, we can extend this sample compression framework to unbounded losses provided that $\mathcal{E}_{\Delta_\lambda}$ is bounded. As an example, we present a result for sub-Gaussian losses (Kahane, 1960).

Corollary 7. *In the setting of Theorem 3, for any $\lambda > 0$, with a ς^2-sub-Gaussian loss function $\ell : \mathcal{H}\times\mathcal{X}\times\mathcal{Y} \to \mathbb{R}$, with probability at least $1-\delta$ over the draw of $S \sim \mathcal{D}^n$, we have*

$$\forall \mathbf{i} \in \mathcal{P}(n), \sigma \in M(\mathbf{i}):$$
$$\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) \leq \widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) + \frac{\lambda\varsigma^2}{2}$$
$$+ \frac{1}{\lambda(n-|\mathbf{i}|)}\left[\log\binom{n}{|\mathbf{i}|} + \log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right].$$

Note that this result encompasses bounded losses with a range of $[a,b]$, as they are sub-Gaussian with $\varsigma = \frac{b-a}{2}$.

It can be extended to the hypothesis-dependent range condition of Haddouche et al. (2021), any unbounded losses under model-dependent assumptions (Casado Telletxea et al., 2025) or more general tail behaviors (Rodríguez-Gálvez et al., 2024).

3.1 Behavior in the consistent case

In this section, we present new theoretical results that justify the tightness of the bounds observed in Section 4, in which we train different types of models with P2L. By construction, P2L is designed to stop when the complement error $\widehat{R}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma))$ is zero. In this consistent setting, where the predictor always finishes training with zero errors, we can prove that Corollaries 4 and 6 are tight upper bounds of Theorem 1.

The first result, presented in Theorem 8, states that the Δ_C bound is an arbitrarily tight upper bound of the binomial tail inversion bound of Laviolette et al. (2005). Indeed, in the following theorem, we show that Corollary 4 is minimized by C tending to ∞ and is equal to Theorem 1 in the limit of $C \to \infty$.

Theorem 8. *In the consistent case, i.e. when $\widehat{R}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) = 0$, Corollary 4 is arbitrarily close to the binomial tail inversion of Theorem 1. Indeed, we have*

$$\overline{\mathrm{Bin}}(0,|\mathbf{i}^c|,\delta^\sigma_{\mathbf{i}}) = \inf_{C>0}\left\{\frac{1-\exp\left(-\frac{1}{|\mathbf{i}^c|}\ln\frac{1}{\delta^\sigma_{\mathbf{i}}}\right)}{1-e^{-C}}\right\} \tag{3}$$
$$= \lim_{C\to\infty}\left\{\frac{1-\exp\left(-\frac{1}{|\mathbf{i}^c|}\ln\frac{1}{\delta^\sigma_{\mathbf{i}}}\right)}{1-e^{-C}}\right\} \tag{4}$$

with $\delta^\sigma_{\mathbf{i}} = \binom{n}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta$.

The following theorem states that the Kullback-Leibler divergence bound is a tight upper bound of the binomial tail inversion bound, up to a constant.

Theorem 9. *In the consistent case, i.e. when $\widehat{R}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)) = 0$, Corollary 6 is a tight upper bound of Theorem 1 up to a constant $K(m,\delta)$. Indeed, we have*

$$\overline{\mathrm{Bin}}(0,|\mathbf{i}^c|,\delta^\sigma_{\mathbf{i}}) \leq \mathrm{kl}^{-1}\left(0, \frac{1}{|\mathbf{i}^c|}\ln\frac{2\sqrt{|\mathbf{i}^c|}}{\delta^\sigma_{\mathbf{i}}}\right) \tag{5}$$
$$= \overline{\mathrm{Bin}}(0,|\mathbf{i}^c|,\delta^\sigma_{\mathbf{i}}) + K(|\mathbf{i}^c|,\delta^\sigma_{\mathbf{i}}), \tag{6}$$

with $K(m,\delta) = \exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right) - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right)$ and $\delta^\sigma_{\mathbf{i}} = \binom{n}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta$.

The proofs of both Theorems 8 and 9 can be found in Appendix C.3.

Note that the constant $K(m,\delta)$ of Equation (6) tends to 0 when m tends to ∞ and is bounded by

$$0 \leq K(m,\delta) \leq \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}.$$

Table 1: Results for the CNNs trained using P2L on the binary MNIST problems. The results displayed obtained the tightest P2L bound. All metrics presented are in percent (%).

Dataset	Validation error	Test error	kl bound	Binomial bound	P2L bound	$\lvert\mathbf{i}\rvert/n$	Baseline test error
MNIST08	0.33±0.17	0.25±0.10	5.05±0.16	5.00±0.16	1.04±0.04	0.87±0.03	0.22±0.05
MNIST17	0.20±0.08	0.38±0.16	4.33±0.21	4.29±0.21	0.86±0.05	0.72±0.04	0.17±0.08
MNIST23	0.39±0.12	0.27±0.10	8.20±0.34	8.15±0.34	1.86±0.09	1.61±0.09	0.16±0.05
MNIST49	0.82±0.11	0.77±0.17	10.52±0.37	10.47±0.37	2.53±0.11	2.23±0.10	0.44±0.08
MNIST56	0.46±0.12	0.47±0.15	6.29±0.22	6.24±0.22	1.35±0.06	1.15±0.05	0.30±0.05

Table 2: Results for the CNNs trained using P2L on the binary MNIST problems and stopped at the iteration with the minimum kl bound. The results displayed obtained the tightest kl bound. Metrics are in percents (%).

Dataset	Validation error	Test error	kl bound	Binomial bound	Train error	$\lvert\mathbf{i}\rvert/n$	Baseline test error
MNIST08	0.49±0.39	0.49±0.26	4.71±0.25	5.33±0.62	0.24±0.23	0.62±0.14	0.22±0.05
MNIST17	0.45±0.18	0.48±0.11	3.70±0.21	4.37±0.11	0.23±0.08	0.43±0.08	0.17±0.08
MNIST23	0.74±0.28	0.84±0.21	6.56±0.38	8.09±0.64	0.64±0.32	0.77±0.20	0.16±0.05
MNIST49	1.16±0.31	1.13±0.24	8.60±0.46	9.61±0.68	0.51±0.28	1.26±0.23	0.44±0.08
MNIST56	0.94±0.09	0.70±0.20	5.42±0.31	6.49±0.81	0.43±0.23	0.65±0.10	0.30±0.05

With $\delta = 0.01$, $K(m,\delta)$ is maximized at $m \approx 7.35$, with $K(7.35, 0.01) \approx 0.11$. However, as depicted by the empirical results of Section 4, $\delta_{\mathbf{i}}^{\sigma}$ is orders of magnitude smaller than 0.01 in practical situations.[2]

4 EXPERIMENTS

In this section, we show the versatility of our results by training different models using the P2L algorithm.[3] In Section 4.1, we train neural networks on binary classification problems and compare our new results to the pre-existing sample compression ones. We empirically validate that our bounds are almost as tight as the binomial bound, all the while not suffering from the numerical optimization problem of Theorem 1 and being defined in the inconsistent case, where the P2L bound of Theorem 2 is undefined. In Section 4.2, we train CNNs on the MNIST dataset and present generalization bounds on the (bounded) cross-entropy loss. As no previous sample-compression bound is defined for real-valued losses, we compare our result to a PAC-Bayesian theorem. In Section 4.3, we use P2L to train tree-based models on regression datasets and give generalization bounds on the root mean squared error (RMSE), an unbounded loss function, under the assumption that it is sub-Gaussian. Finally, in Section 4.4, we fine-tune DistilBERT, a 66M parameters language model, on a review polarity classification problem. We obtain tight bounds simultaneously on the zero-one loss and the cross-entropy loss, demonstrating that our new theorem is independent of the number of parameters of the model.

Each experiment is run five times with different seeds. We report the mean and standard deviation of the metrics over these five repetitions. The datasets are separated into three parts: the training, validation and test set. When a dataset doesn't have a built-in test set, we create it using 10% of the samples. Of the remaining samples, 10% are used for the validation set size and 90% for the training set. When computing the bounds, we use $\delta = 0.01$. The hyperparameters used for the experiments can be found in Appendix A.

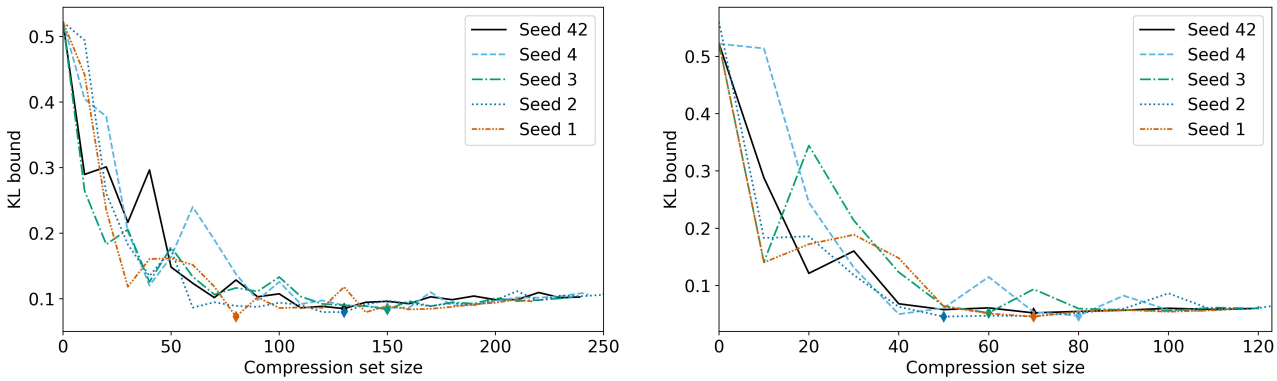
4.1 Binary MNIST

We create binary classification datasets by extracting pairs of digits from the MNIST dataset (LeCun et al., 1998), e.g., selecting the datapoints labeled 0 and 8 to build the dataset MNIST08. We create five datasets: MNIST08, MNIST17, MNIST23, MNIST49 and MNIST56. Starting from randomly initialized neural networks, we train a MLP and a CNN using P2L on each dataset. More details are given in Appendix A.1.1.

For all experiments in this section, we compute our proposed kl bound (Corollary 6), the binomial approximation bound of Laviolette et al. (2005) (Corollary 10, in appendix) and the P2L bound of Paccagnan et al. (2024) (Theorem 2). We do not compute the binomial tail inversion of Theorem 1 as its optimization is very unstable. However, the binomial approximation is equivalent to Theorem 1 when $k = 0$, which corresponds to the consistent case reached by the P2L algorithm.

[2] Given a dataset of 10597 datapoints, a compression set of size 92 and $\delta = 0.01$, we have $K(10505, 10^{-234}) \approx 0.0005$. This is indeed the difference that we observe in Table 1 between the kl bound and the binomial bound for MNIST08.

[3] Our code is available at `https://github.com/GRAAL-Research/pick-to-learn`.



(a) MNIST49 (b) MNIST56

Figure 1: Illustration of the behavior of the kl bound throughout P2L iterations for the five different seeds of the hyperparameter combination that achieved the minimal P2L bound on MNIST49 and MNIST56. We mark the minimal kl bound for each seed with a diamond (⧫). The results for the other datasets can be found in Fig. 2.

Table 3: Cross-entropy loss achieved by the CNNs on MNIST. The results displayed obtained the smallest kl bound.

Learning algorithm	Train loss	Test loss	kl bound	$\lvert\mathbf{i}\rvert/n$	Baseline test loss
P2L	0.0014±0.0006	0.0467±0.0056	0.8279±0.0256	1.13±0.24	0.0499±0.0042
PBB	0.0092±0.0005	0.0045±0.0004	0.0112±0.0005	-	

Table 1 presents results for the CNN in the consistent case. The results for the MLP can be found in Appendix A.1.1. The error on the training set is zero for all predictors returned by P2L. The presented results achieve the tightest P2L bound for each dataset. The reported "baseline test error" corresponds to the results of a standard neural network optimized on the whole training set by stochastic gradient descent for 200 epochs or until the model achieves zero training errors; the selected hyperparmeters are the ones minimizing the validation error. For both CNN and MLP architectures, using P2L only incurs a slight increase of the test error compared to the baseline, whilst the model is trained on a fraction of the dataset, ranging from 0.7% to 3.4%. Finally, even though the P2L bound is much tighter than the proposed kl bound, our result is much more general, as it holds for any real-valued loss functions and in the non-consistent case. Moreover, our bounds hold uniformly over all iterations of the models trained using P2L. After training, one can use any checkpoint of the model and still obtain a valid bound, which gives control over a trade-off between the training error, the generalization bound and the validation error.

As for the inconsistent case, Figure 1 presents the behavior of the bound throughout the P2L iterations. We observe that the minimal kl bound value happens at about half the final number of iterations, leading to a smaller compression set and a tighter bound, as also reported in Table 2. Recall that the P2L bound cannot be computed in this case, as the model do not reach zero errors. In comparison to the previous consistent results (Table 1), the test error of Table 2 are about twice as high as the fully trained model. However, the inconsistent models are trained on very small portions of the dataset, with the MNIST17 model being trained on 0.42% of the dataset and still achieving a test error of 0.48%. Finally, we observe that, in this setting, our new kl bound is much tighter than the binomial approximation of Laviolette et al. (2005).

4.2 MNIST

We now train convolutional neural networks composed of two convolutional layers and two fully connected layers. We pre-train the model using stochastic gradient descent on a subset of the dataset and then use P2L to fine-tune the model on the train set. The size of the pre-training subset is an hyperparameter. We use the same training setting as in Section 4.1 and use the extension of P2L that adds multiple datapoints to the compression set at a time, with batch size $R = 32$, as defined by Algorithm 2 of Paccagnan et al. (2024). For comparison, we also train probabilistic neural networks (PNN) using the PAC-Bayes with Backprop (PBB) approach of Pérez-Ortiz et al. (2021), which optimize a PAC-Bayesian kl bound (Theorem 11 in appendix).

For both our new sample-compression bounds and the PAC-Bayesian bound of Pérez-Ortiz et al. (2021), we compute the bounds on the zero-one loss and on a bounded version of the cross-entropy loss (see Ap-

Table 4: Results for the decision forests trained using P2L. We report the RMSE achieved by the models with the smallest kl bound. The ratio $|\mathbf{i}|/n$ is presented in percents (%).

Dataset	Train loss	Validation loss	Test loss	kl bound	Linear bound	$\lvert\mathbf{i}\rvert/n$	Baseline test loss	$\ell^{\max}$
Powerplant	6.11±0.89	6.23±0.70	6.31±0.95	13.69±0.27	15.92±0.47	0.51±0.17	3.59±0.13	90.6
Infrared	0.27±0.03	0.29±0.04	0.30±0.03	1.08±0.08	1.16±0.08	2.32±0.66	0.23±0.01	4.26
Airfoil	3.67±0.16	4.03±0.37	3.90±0.18	14.19±0.49	14.25±0.42	3.23±0.46	2.10±0.15	45.13
Parkinson	7.79±0.33	7.79±0.28	7.84±0.27	12.24±0.27	12.02±0.23	0.43±0.10	2.23±0.16	41.37
Concrete	8.18±0.91	8.70±1.00	8.48±1.41	31.68±1.63	32.49±1.52	3.81±0.82	4.70±0.36	90.63

Table 5: Results for the decision trees trained using P2L. We report the RMSE achieved by the models with the smallest kl bound. The ratio $|\mathbf{i}|/n$ is presented in percents (%)

Dataset	Train loss	Validation loss	Test loss	kl bound	Linear bound	$\lvert\mathbf{i}\rvert/n$	Baseline test loss	$\ell^{\max}$
Powerplant	11.66±3.00	11.83±3.12	12.00±3.04	23.40±1.59	24.15±1.54	0.94±0.49	4.07±0.13	90.6
Infrared	0.48±0.10	0.47±0.11	0.48±0.07	1.34±0.03	1.33±0.06	2.10±1.06	0.27±0.03	4.26
Airfoil	11.02±1.71	10.89±1.38	11.10±1.93	18.87±1.97	18.14±1.75	1.00±0.03	3.01±0.19	45.13
Parkinson	13.93±2.83	13.84±2.76	13.99±2.92	17.75±3.34	15.10±7.23	0.25±0.00	3.20±0.15	41.37
Concrete	26.08±4.96	25.05±4.38	26.40±4.05	45.79±4.26	44.47±4.24	1.51±0.33	6.22±0.91	90.63

pendix A.1.2). The probabilities outputted by the neural networks are restricted to be greater than 10^{-5}, effectively bounding the cross-entropy by $-\ln\big(10^{-5}\big) \approx 11.51$.

Table 3 reports the bound values for the bounded cross-entropy loss (see Table 11 for classification error). We observe that the PBB algorithm gives a tighter generalization bound than the one of P2L. This gap can be explained by the fact that PBB jointly optimizes the train error and the KL divergence, whilst we have almost no control on the minimization of the bound. Indeed, the heuristic of the P2L algorithm, which is to choose the datapoints over which the model incurs the greatest losses, doesn't give control on the trade-off between the decrease of the error and the increase of the complexity term. Moreover, for a large dataset, the binomial coefficient increases rapidly when the compression set size increases. However, using our bounds with the P2L algorithm has multiple advantages over the PBB algorithm. First of all, PBB needs to train twice as many parameters, as it fits both the mean and standard deviation of the distributions over the parameters. Secondly, computing the PAC-Bayesian bound necessitates a step of Monte Carlo sampling to determine the average error of the model. For 5000 steps of Monte Carlo sampling, the prediction over the dataset is computed 5000 times, instead of only once with P2L. Finally, our bound doesn't take into account the number of parameters of the model, whilst the KL divergence in Theorem 11 is a sum of the KL divergence of the distribution of each parameter of the model.

4.3 Regression with tree-based models

In order to show the wide applicability of our bounds, we train decision forests on regression problems: Powerplant (Tüfekci, 2014), Infrared (Wang et al., 2021), Airfoil (Brooks et al., 1989b), Parkinson (Tsanas et al., 2009) and Concrete (Yeh, 1998b). These datasets range from a training set size of 827 to 7751 and range from a number of features of 4 to 33. To the best of our knowledge, no sample compression bounds exist for this setting. Previous results were presented for linear regression under an ℓ_p loss (Attias et al., 2018, 2023, 2024) or for boosting real-valued learner in a binary classification setting (Hanneke et al., 2019), none of which are equivalent to our setting. We adapt the P2L algorithm to this regression problem (see Algorithm 2 in appendix), which differs from the original one, designed only for classification problems where zero training error is achievable (consistent case). At each P2L iteration, we add a single datapoint to the compression set in order to train the forest. The selected datapoint is the one with the largest root mean squared error (RMSE). Then, the trees are retrained from scratch on the compression set. As the minimal RMSE that can be achieved is dependent on the dataset, setting a predetermined threshold is not a suitable stopping criterion. Thus, we train the model until the validation loss has not decreased for a given number of iterations. To compute the bounds, we need the loss to be either bounded or sub-Gaussian. As tree-based models predict the mean of the targets of each datapoint assigned to a leaf, their outputs are bounded by the extrema of the data. To compute the kl bound, we assume that the target space is bounded by the maximum value of the loss $\ell^{\max}$ reported in Table 4. To compute the linear bound, we assume that the loss is sub-Gaussian. We discuss in more details these assumptions and the way of defining the extrema in Appendix A.1.3.

Table 4 contains the results of the models selected based

Table 6: Results for the amazon polarity dataset. The results displayed for P2L obtained the lowest kl bound on the error, whilst the baseline was chosen by the lowest validation error. The ratio $|\mathbf{i}|/n$ is presented in percents (%).

Learning algorithm	Error (%)				Cross-entropy loss			
	Train	Test	kl bound	Binomial bound	Train	Test	kl bound	$\lvert\mathbf{i}\rvert/n$
P2L	4.73±1.09	5.60±1.19	13.91±2.73	21.85±3.29	0.1199±0.0118	0.1478±0.0182	0.8594±0.1622	0.79±0.23
Baseline	3.11±0.02	4.19±0.00	-	-	0.0912±0.0010	0.1158±0.0002	-	-

on the smallest kl bounds. We observe that the models trained with P2L are able to obtain competitive results with respect to the test error of standard random forests trained on the whole dataset. We report latter results in the column "baseline test loss", where the models are chosen by their validation loss. As the values of the bounds are much smaller than $\ell^{\max}$, we conclude that our bounds are tight and non-vacuous. The generalization guarantees given by the bounds relying on the linear function are competitive to the ones relying on the kl; they are even tighter on the Parkinson dataset.

Following a similar experiment setting, we trained regression trees with P2L. As we can see in Table 5, training trees using P2L leads to underfitted models that are not competitive with respect to the baseline. Indeed, as the trees are trained on a small subset of the data, they are restricted to be less complex than trees trained on the whole dataset. When selecting the models by the smallest validation loss (see Table 13), we observe that the models achieve better performance, as the compression sets are much larger, but also suffer from worsened bounds.

4.4 Amazon polarity

Finally, we train DistilBERT (Sanh et al., 2019) on the Amazon reviews polarity dataset (Zhang et al., 2015). Using P2L, we fine-tune the pretrained language model on 10% of the dataset, for a total of 360k datapoints, and evaluate the model on the test set, which comprises 400k datapoints. We pre-train the model on half of the training dataset and then use P2L on the other half of the training set. We add 32 datapoints at a time in the compression set and early stop the training of the model if its validation loss has not decreased for 20 epochs. In this experiment, we study our new kl bound on the zero-one loss and on the bounded cross-entropy loss. Moreover, we compute the binomial approximation bound of Corollary 10. The P2L bound (Theorem 2) is inapplicable in this setting, as the model doesn't reach zero errors. The PAC-Bayesian bound of Theorem 11 could be computed on both metrics, but it would necessitate to train 132M parameters (twice the number of parameters of DistilBERT). Many new generalization bounds and approaches were presented for very large models (Lotfi et al., 2024, 2025; Zekri et al., 2024; Su et al., 2025), such as large language models. However, most approaches are not suited for classification and regression, as they are derived for language modeling objectives.

From the results displayed by Table 6, we first observe that training the model using P2L only incurs a loss of about a percent for the train, validation and test error compared to the baseline. It achieves this error whilst being trained on about 0.8% of the dataset, as the compression set size is roughly 1138 datapoints and the training set size is 144k. Both for the error and the cross-entropy loss, the bound is tight and non-vacuous. Our bound is much tighter than the binomial approximation bound, with a certificate of 13.91% for a train error of 4.73%. Thus, despite the 66M parameters of DistilBERT, we are able to obtain tight generalization guarantees by simply changing the training loop of the model for the P2L scheme.

5 CONCLUSION

We proposed novel generalization bounds for real-valued losses and sample-compressed predictors. These bounds leverage the comparator functions studied in the PAC-Bayes theory. We provide results for bounded and unbounded losses, under different assumptions. We empirically verified the tightness of the proposed bounds, showing that it is almost as tight as the binomial tail inversion, which, however, holds only for a less general setting. We trained neural networks with 66M parameters and obtained tight guarantees, without suffering from the cost of the number of parameters. This highlights an important asset of the sample compression framework: Two models achieving the same empirical loss using the same amount of datapoints (compression set size) share the same guarantees (bound value), regardless of their size in terms of the number of trainable parameters.

In future works, we could leverage the possibility of having a message in the compression scheme, by training models such as the set covering machine (Laviolette et al., 2005) or decision trees (Shah, 2007), which both use binary sequences to specify how to reconstruct the model. Finally, although P2L is generally able to train good performing models, it is unclear that its sample selection heuristic is optimal for neural networks. Trying different heuristics, e.g., that optimize for sample diversity, could lead to further improvements.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. We also wish to thank Benjamin Leblanc and Sokhna Diarra Mbacke for the insightful discussions and their help proof-reading the manuscript.

Mathieu Bazinet is supported by a FRQNT B2X scholarship (343192). Pascal Germain is supported by the Canada CIFAR AI Chair Program and the NSERC Discovery grant RGPIN-2020-07223.

References

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P., and Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM.

Attias, I., Hanneke, S., Kalavasis, A., Karbasi, A., and Velegkas, G. (2023). Optimal learners for realizable regression: Pac learning and online learning. *Advances in Neural Information Processing Systems*, 36:44707–44739.

Attias, I., Hanneke, S., Kontorovich, A., and Sadigurschi, M. (2018). Agnostic sample compression schemes for regression. In *Forty-first International Conference on Machine Learning*.

Attias, I., Hanneke, S., and Ramaswami, A. (2024). Sample compression scheme reductions. *ArXiv preprint*, abs/2410.13012.

Ben-David, S., Bie, A., Canonne, C. L., Kamath, G., and Singhal, V. (2024). Private distribution learning with public data: The view from sample compression. *Advances in Neural Information Processing Systems*, 36.

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Brooks, T. F., Pope, D. S., and Marcolini, M. A. (1989a). Airfoil Self-Noise. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5VW2C.

Brooks, T. F., Pope, D. S., and Marcolini, M. A. (1989b). Airfoil self-noise and prediction. Technical report.

Campi, M. C. and Garatti, S. (2023). Compression, generalization and learning. *Journal of Machine Learning Research*, 24(339):1–74.

Casado Telletxea, I., Ortega Andrés, L. A., Pérez, A., and Masegosa, A. (2025). Pac-bayes-chernoff bounds for unbounded losses. *Advances in Neural Information Processing Systems*, 37:24350–24374.

Catoni, O. (2007). Pac-bayesian supervised classification: the thermodynamics of statistical learning. *ArXiv preprint*, abs/0712.0248.

David, O., Moran, S., and Yehudayoff, A. (2016). Supervised learning through the lens of compression. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2784–2792.

Drouin, A., Letarte, G., Raymond, F., Marchand, M., Corbeil, J., and Laviolette, F. (2019). Interpretable genotype-to-phenotype classifiers with performance guarantees. *Scientific Reports*, 9(11):4071.

Dziugaite, G. K. and Roy, D. M. (2018). Data-dependent pac-bayes priors via differential privacy. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8440–8450.

Falcon, W. and The PyTorch Lightning team (2019). PyTorch Lightning.

Floyd, S. and Warmuth, M. (1995). Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine learning*, 21(3):269–304.

Foong, A., Bruinsma, W., Burt, D., and Turner, R. (2021). How tight can pac-bayes be in the small data regime? *Advances in Neural Information Processing Systems*, 34:4093–4105.

Germain, P., Lacasse, A., Laviolette, F., and Marchand, M. (2009). Pac-bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 353–360. ACM.

Germain, P., Lacasse, A., Laviolette, F., Marchand, M., and Roy, J.-F. (2015). Risk bounds for the majority vote: From a pac-bayesian analysis to a learning algorithm. *The Journal of Machine Learning Research*, 16:787–860.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial*

intelligence and statistics, pages 315–323. JMLR Workshop and Conference Proceedings.

Graepel, T., Herbrich, R., and Shawe-Taylor, J. (2005). PAC-Bayesian Compression Bounds on the Prediction Error of Learning Algorithms for Classification. *Machine Learning*, 59(1):55–76.

Haddouche, M., Guedj, B., Rivasplata, O., and Shawe-Taylor, J. (2021). Pac-bayes unleashed: Generalisation bounds with unbounded losses. *Entropy*, 23(10):1330.

Hanneke, S. and Kontorovich, A. (2021). Stable sample compression schemes: New applications and an optimal SVM margin bound. In *Algorithmic Learning Theory*, pages 697–721. PMLR.

Hanneke, S., Kontorovich, A., and Sadigurschi, M. (2018). Efficient Conversion of Learners to Bounded Sample Compressors. *Proceedings of Machine Learning Research vol*, 75:1–21.

Hanneke, S., Kontorovich, A., and Sadigurschi, M. (2019). Sample Compression for Real-Valued Learners. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pages 466–488. PMLR.

Hanneke, S., Moran, S., and Tom, W. (2024). List sample compression and uniform convergence. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 2360–2388. PMLR.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hellström, F. and Guedj, B. (2024). Comparing comparators in generalization bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 73–81. PMLR.

Hussain, Z., Laviolette, F., Marchand, M., Shawe-Taylor, J., Brubaker, S. C., and Mullin, M. D. (2007). Revised loss bounds for the set covering machine and sample-compression loss bounds for imbalanced data. *Journal of Machine Learning Research*, 8(84):2533–2549.

Kahane, J. (1960). Propriétés locales des fonctions à séries de fourier aléatoires. *Studia Mathematica*, 19(1):1–25.

Kestler, H. A., Lindner, W., and Müller, A. (2006). Learning and feature selection using the set covering machine with data-dependent rays on gene expression profiles. In *Artificial Neural Networks in Pattern Recognition: Second IAPR Workshop, ANNPR 2006, Ulm, Germany, August 31-September 2, 2006. Proceedings 2*, pages 286–297. Springer.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Langford, J. (2005). Tutorial on practical prediction theory for classification. *Journal of machine learning research*, 6(3).

Langford, J. and Seeger, M. (2001). *Bounds for averaging classifiers*. School of Computer Science, Carnegie Mellon University.

Laviolette, F., Marchand, M., and Shah, M. (2005). Margin-Sparsity Trade-Off for the Set Covering Machine. In *Machine Learning: ECML 2005*, volume 3720, pages 206–217. Springer Berlin Heidelberg.

Laviolette, F., Marchand, M., Shah, M., and Shanian, S. (2009). Learning the set covering machine by bound minimization and margin-sparsity trade-off. *Machine Learning*, 78:175–201.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Letarte, G., Germain, P., Guedj, B., and Laviolette, F. (2019). Dichotomize and generalize: Pac-bayesian binary activated deep neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6869–6879.

Littlestone, N. and Warmuth, M. (1986). Relating data compression and learnability.

Lotfi, S., Finzi, M. A., Kuang, Y., Rudner, T. G. J., Goldblum, M., and Wilson, A. G. (2024). Non-vacuous generalization bounds for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32801–32818. PMLR.

Lotfi, S., Kuang, Y., Finzi, M., Amos, B., Goldblum, M., and Wilson, A. G. (2025). Unlocking tokens as data points for generalization bounds on larger language models. *Advances in Neural Information Processing Systems*, 37:9229–9256.

Marchand, M., Shah, M., Shawe-Taylor, J., and Sokolova, M. (2003). The set covering machine with data-dependent half-spaces. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 520–527. AAAI Press.

Marchand, M. and Shawe-Taylor, J. (2002). The set covering machine. *Journal of Machine Learning Research*, 3(4-5):723–746.

Marchand, M. and Sokolova, M. (2005). Learning with decision lists of data-dependent features. *Journal of Machine Learning Research*, 6(4).

Maurer, A. (2004). A note on the pac bayesian theorem. *arXiv preprint cs/0411099*.

McAllester, D. A. (1998). Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234.

Moran, S., Nachum, I., Panasoff, I., and Yehudayoff, A. (2020). On the perceptron's compression. In *Beyond the Horizon of Computability: 16th Conference on Computability in Europe, CiE 2020, Fisciano, Italy, June 29–July 3, 2020, Proceedings 16*, pages 310–325. Springer.

Moran, S. and Yehudayoff, A. (2016). Sample compression schemes for vc classes. *Journal of the ACM (JACM)*, 63(3):1–10.

Paccagnan, D., Campi, M., and Garatti, S. (2024). The pick-to-learn algorithm: Empowering compression for tight generalization bounds and improved post-training performance. *Advances in Neural Information Processing Systems*, 36.

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

Pérez-Ortiz, M., Rivasplata, O., Shawe-Taylor, J., and Szepesvári, C. (2021). Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22(227):1–40.

Rodríguez-Gálvez, B., Thobaben, R., and Skoglund, M. (2024). More pac-bayes bounds: From bounded losses, to losses with general tail behaviors, to anytime validity. *Journal of Machine Learning Research*, 25(110):1–43.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rubinstein, B. I. and Rubinstein, J. H. (2012). A geometric approach to sample compression. *Journal of Machine Learning Research*, 13(4).

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv preprint*, abs/1910.01108.

Seeger, M. (2002). Pac-bayesian generalisation error bounds for gaussian process classification. *Journal of machine learning research*, 3(Oct):233–269.

Shah, M. (2007). Sample compression bounds for decision trees. In *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.

Shah, M., Marchand, M., and Corbeil, J. (2011). Feature selection with conjunctions of decision stumps and learning from microarray data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):174–186.

Snyder, C. and Vishwanath, S. (2020). Sample compression, support vectors, and generalization in deep learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1):106–120.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Su, J., Kempe, J., and Ullrich, K. (2025). Mission impossible: A statistical perspective on jailbreaking llms. *Advances in Neural Information Processing Systems*, 37:38267–38306.

Tfekci, P. and Kaya, H. (2014). Combined Cycle Power Plant. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5002N.

Tsanas, A. and Little, M. (2009). Parkinsons Telemonitoring. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5ZS3N.

Tsanas, A., Little, M., McSharry, P., and Ramig, L. (2009). Accurate telemonitoring of parkinson's disease progression by non-invasive speech tests. *Nature Precedings*, pages 1–1.

Tüfekci, P. (2014). Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power & Energy Systems*, 60:126–140.

Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280.

Wang, Q., Zhou, Y., Ghassemi, P., Chenna, D., Chen, M., Casamento, J., Pfefer, J., and Mcbride, D. (2023). Facial and oral temperature data from a large set of human subject volunteers.

Wang, Q., Zhou, Y., Ghassemi, P., McBride, D., Casamento, J. P., and Pfefer, T. J. (2021). Infrared

thermography for measuring elevated body temperature: clinical accuracy, calibration, and evaluation. *Sensors*, 22(1):215.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Yeh, I.-C. (1998a). Concrete Compressive Strength. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PK67.

Yeh, I.-C. (1998b). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808.

Zekri, O., Odonnat, A., Benechehab, A., Bleistein, L., Boullé, N., and Redko, I. (2024). Large language models as markov chains. *ArXiv preprint*, abs/2410.02724.

Zhang, X., Zhao, J. J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes, in Section 2.**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Not applicable.**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes, the code can be found here :** `https://github.com/GRAAL-Research/pick-to-learn`**.**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes, the only assumption is that the data is *i.i.d.* and is mentioned in our setting in Section 2.**
   (b) Complete proofs of all theoretical results. **Yes, in Appendix C.**
   (c) Clear explanations of any assumptions. **The only assumption is that the data is *i.i.d.*, which is the classical setting in the literature.**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **All the code and instruction can be found in the repository. The data is imported directly when the code is executed.**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **All the training details can be found in Appendix A and the code.**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **All metrics are defined in either Section 4 or Appendix A. We present the mean and standard deviation over five seeds.**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes, in Appendix A.**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes.**
   (b) The license information of the assets, if applicable. **Yes, in Appendix A.**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Yes, the code can be found here :** `https://github.com/GRAAL-Research/pick-to-learn`**.**
   (d) Information about consent from data providers/curators. **Not applicable.**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not applicable.**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not applicable**

Sample compression unleashed : Supplementary Materials

A EXPERIMENTS

Devices. The experiments were run on two different devices. The experiments with PBB algorithm and the regression datasets were run on Python 3.12.2 on a computer with a NVIDIA GeForce RTX 4090. The experiments on MNIST were run on Python 3.12.3 on a computer with a NVIDIA GeForce RTX 2080 Ti.

Librairies. The libraries used for each environment can be found with the code. Notably, we use PyTorch (Ansel et al., 2024) (BSD 3-Clause License), Lightning (Falcon and The PyTorch Lightning team, 2019) (Apache 2.0 license), Weight and Biases (Biewald, 2020) (MIT License), Scikit-Learn (Pedregosa et al., 2011) (BSD 3-Clause License), NumPy (Harris et al., 2020) (NumPy license) and Transformer (Wolf et al., 2020) (Apache 2.0 license). For all experiments, we run the code with the following seeds : {1,2,3,4,42}.

Datasets. For the classification problems, we use the MNIST dataset (LeCun et al., 1998) (MIT License) and the amazon polarity dataset (Zhang et al., 2015) (Apache 2.0 License). All MNIST derived-dataset are composed of 784 real-valued features. For the multi-class classification problems on MNIST, we denote MNIST (p%) to say that we pre-train the model on p% of the data, where p is a hyperparameter. For the Amazon polarity dataset, we chose 10% of the dataset to create a 360k datapoints dataset. We then use 50% to pre-train the model and split the rest into a training and validation set. The datapoints are textual reviews and the labels are binary. The description of the dataset are presented in Table 7.

Table 7: Description of the datasets used for classification problems.

Dataset	Pretrain set size	Train set size	Validation set size	Test set size
Amazon Polarity	180000	144000	36000	400000
MNIST (10%)	6000	48000	6000	10000
MNIST (20%)	12000	42000	6000	10000
MNIST (50%)	30000	24000	6000	10000
MNIST08	0	10597	1177	1954
MNIST17	0	11707	1300	2163
MNIST23	0	10881	1208	2042
MNIST49	0	10612	1179	1991
MNIST56	0	10206	1133	1850

For the regression problems, we train our models on five datasets : the *Combined Cycle Power Plant* (Tüfekci, 2014; Tfekci and Kaya, 2014), the *Infrared Thermography Temperature* (Wang et al., 2021, 2023), the *Airfoil Self-Noise* (Brooks et al., 1989b,a), the *Parkinsons Telemonitoring* (Tsanas et al., 2009; Tsanas and Little, 2009) and the *Concrete Compressive Strength* (Yeh, 1998b,a). The descriptions of the dataset are presented in Table 8. All datasets were chosen from the UCI dataset repository. Powerplant, Airfoil, Parkinson and Concrete are under the CC-BY 4.0 license. The Infrared dataset is under the CC0 license.

Table 8: Description of the datasets used for regression problems.

Dataset	Train set size	Validation set size	Test set size	Number of features
Powerplant	7751	861	956	4
Infrared	827	91	102	33
Airfoil	1218	135	150	5
Parkinson	4760	528	587	19
Concrete	835	92	103	8

A.1 Hyperparameter grids

In this section, we present the hyperparameter grids for all the experiments.

In all experiments, we use $\delta = 0.01$ and a batch size of 64. After each iteration of P2L, we train the model for 200 epochs or until the validation loss has not improved for three epochs.

A.1.1 Binary MNIST problems

For the binary MNIST problems, we used the following hyperparameters for both MLP and CNN architectures.

- Dropout probability : $\{0.1, 0.2\}$
- Training learning rate : $\{10^{-2}, 10^{-3}, 5\times10^{-3}, 10^{-4}\}$

The MLP is composed of three hidden fully connected layers of 600 neurons and the CNN is composed of two convolutional layers and two fully connected layers. We use ReLU activations (Glorot et al., 2011), dropout layers (Srivastava et al., 2014) and the Adam optimizer (Kingma and Ba, 2015) with the default parameters $\beta = (0.9, 0.999)$.

At each iteration, the P2L algorithm adds one datapoint to the compression set. We use h_0 a randomly initialized neural network.

For the baselines, we train the same models with the same hyperparameters for 200 epochs or until the model achieves zero errors on the training set.

We present the results for the MLP, both trained fully using P2L and early-stopped, respectively in Table 9 and in Table 10. Moreover, Fig. 2 displays the results not present in Fig. 1.

Table 9: Results for the MLPs trained using P2L on the binary MNIST problems. The results displayed obtained the tightest P2L bound. All metrics presented are in percents (%).

Dataset	Validation error	Test error	kl bound	Binomial bound	P2L bound	$\lvert\mathbf{i}\rvert/n$	Baseline test error
MNIST08	0.41±0.14	0.40±0.08	6.56±0.30	6.51±0.30	1.42±0.08	1.21±0.07	0.34±0.07
MNIST17	0.37±0.14	0.47±0.17	4.93±0.27	4.89±0.27	1.01±0.07	0.85±0.06	0.33±0.09
MNIST23	0.87±0.24	0.58±0.12	12.21±0.29	12.17±0.29	3.06±0.09	2.73±0.09	0.36±0.14
MNIST49	1.19±0.33	1.04±0.10	14.41±0.05	14.37±0.05	3.78±0.02	3.41±0.02	0.96±0.15
MNIST56	0.68±0.17	0.65±0.05	10.35±0.31	10.30±0.31	2.48±0.09	2.18±0.09	0.59±0.01

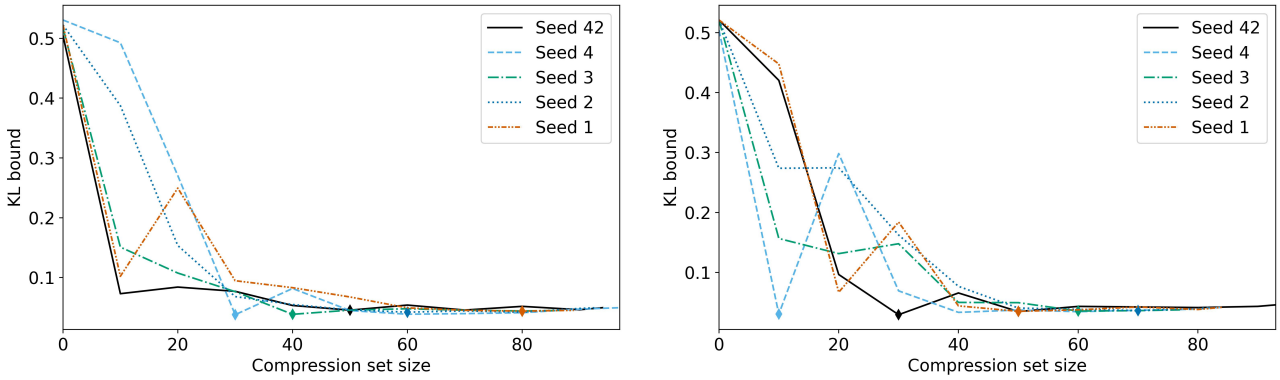
A.1.2 MNIST problems

We train a convolutional neural network over the 10-class MNIST dataset with the following hyperparameters.

- Size of pretraining set : $\{10\%, 20\%, 50\%\}$
- Pretraining epochs : $\{50, 100\}$
- Pretraining learning rate : $\{10^{-2}, 10^{-3}, 10^{-4}\}$
- Dropout probability : $\{0.1, 0.2\}$
- Training learning rate : $\{10^{-2}, 5\times10^{-3}, 10^{-4}\}$

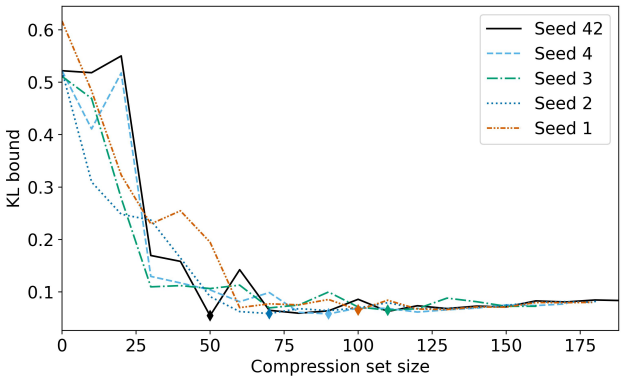
Table 10: Results for the MLPs trained using P2L on the binary MNIST problems and stopped at the iteration with the minimum kl bound. The results displayed obtained the tightest kl bound. All metrics presented are in percents (%).

Dataset	Validation error	Test error	kl bound	Binomial bound	Train error	$\lvert\mathbf{i}\rvert/n$	Baseline test error
MNIST08	1.11±0.52	1.04±0.67	5.46±0.53	7.77±1.64	0.85±0.71	0.56±0.32	0.34±0.07
MNIST17	0.88±0.39	0.80±0.29	4.02±0.36	5.49±0.77	0.50±0.26	0.38±0.13	0.33±0.09
MNIST23	1.93±0.49	1.59±0.43	10.86±0.19	13.23±0.74	1.27±0.41	1.34±0.24	0.36±0.14
MNIST49	2.28±0.53	2.07±0.58	13.14±0.32	15.08±0.99	1.22±0.47	1.90±0.29	0.96±0.15
MNIST56	1.97±0.53	1.88±0.44	8.85±0.58	11.78±1.44	1.38±0.61	0.90±0.27	0.59±0.01



(a) MNIST08

(b) MNIST17



(c) MNIST23

Figure 2: Illustration of the behavior of the kl bound throughout P2L iterations for the five different random seed initializations sharing the hyperparameter combination that achieved the minimal averaged P2L bound. The diamonds (⧫) mark the minimal kl bound for each run.

At each iteration, the P2L algorithm adds 32 datapoints to the compression set. We use h_0 a neural network pre-trained on $p\%$ of the dataset (see size of pretraining set for the different values of p).

To compute bounds for the cross-entropy loss, we clamp the log-probabilities to be greater or equal than $\ln(10^{-5})$ (Pérez-Ortiz et al., 2021; Dziugaite and Roy, 2018), as follows :

$$\ell(h,\boldsymbol{x},y) = -\max\left(\ln(10^{-5}),\ln\left(\frac{\exp(h(\boldsymbol{x})_y)}{\sum_{c=1}^{C}\exp(h(\boldsymbol{x})_c)}\right)\right),$$

where $h(\boldsymbol{x}) = (h(\boldsymbol{x})_1,\ldots,h(\boldsymbol{x})_C)$ is the output of the neural network and C is the number of classes. The loss then takes values in $[0,-\ln(10^{-5})]$. We use the same bounded cross-entropy loss for the following experiments.

Table 11: Classification risk achieved by the CNNs on MNIST. The results displayed obtained the smallest kl bound. All metrics are presented in percents (%).

Learning algorithm	Train error	Test Error	kl bound	Binomial bound	$\lvert\mathbf{i}\rvert/n$	Baseline test error
P2L	0.0±0.0	1.14±0.07	7.11±0.22	7.08±0.22	1.15±0.34	1.08±0.09
PBB	1.67 ± 0.07	1.05±0.05	1.94±0.07	-	-	

For the baseline, we train the same model with the same hyperparameters for 200 epochs or until the model achieves zero errors on the training set.

For the PAC-Bayes with Backprop (PBB) algorithm, we used the code of the GitHub repository provided alongside Pérez-Ortiz et al. (2021), with the same hyperparameter grid proposed by the authors,[4] except for the dropout rate, which we kept the same as the other experiments:

- Scale parameter of the prior distribution : {0.1,0.05,0.04,0.03,0.02,0.01,0.005}
- Training learning rate : $\{10^{-3},5\times10^{-3},10^{-2}\}$
- Pre-training learning rate : $\{10^{-3},5\times10^{-3},10^{-2}\}$
- Momentum : {0.95,0.99}
- Dropout probability : {0.1,0.2}

We fixed $\delta=\delta'=0.01$ to compute Theorem 11, and performed $m=5000$ Monte Carlo sampling steps instead of using the value $m=150000$ found in the code, as it takes several hours to run.

A.1.3 Regression problems

We trained decision trees and forests on the datasets, using P2L to train the models on one datapoint at a time. We trained the models until their validation loss hasn't decreased for 10 or 20 epochs. We summarize this idea in Algorithm 2. We denote the RMSE as $\ell^{\mathrm{RMSE}}(h,\boldsymbol{x},y)=\sqrt{(h(\boldsymbol{x})-y)^2}$ and the empirical risk on the dataset

$$\mathcal{L}_S^{\mathrm{RMSE}}(h)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(h(\boldsymbol{x}_i)-y_i)^2}.$$

For tree-based models, we chose h_0 to simply output zeroes for all entries. We use `COUNTER` and $\widehat{\mathcal{L}}_{\mathrm{BEST}}$ as variables to stop the training when the loss hasn't decreased for T epochs.

We now present the hyperparameter grid.

- Maximum depth of the trees : {5,10}
- Minimum samples to split : {2,3,4}
- Minimum samples to create a leaf : {1,2,3}
- Cost-Complexity pruning parameter : {0.0,0.05,0.1,0.2,0.5,1,2}
- Number of epochs before early stopping: {10,20}

For the decision forests, we choose the number of estimators in {50,100}. For the baselines, we train the same model with the same hyperparameters on the whole dataset. We present the results for the forests in Table 4 and Table 12.

The results for the decision trees can be found in Table 5 and Table 13. Using only P2L to train the trees leads to underfitted trees, as the model is not complex enough to use only a few datapoints to train a complete model.

When computing the generalization bounds, we need to assume that the loss is bounded (for the kl bound) or sub-gaussian (for the linear bound).

Note that the RMSE is not bounded. However, the regression trees cannot predict a value greater (respectively lower) than the highest (respectively lowest) target value found in the training dataset. To compute the kl bound, we work under the assumption that the target space $\mathcal{Y}$ is such that

$$\mathcal{Y}=\left[y_- - \tfrac{1}{10}(y_+-y_-), y_+ - \tfrac{1}{10}(y_+-y_-)\right] \text{ with } y_-=\min_{(\boldsymbol{x},y)\in S} y \text{ and } y_+=\max_{(\boldsymbol{x},y)\in S} y.$$

[4] `https://github.com/mperezortiz/PBB`

Algorithm 2: Pick-To-Learn for regression problems

Input : T, the number of look-ahead iterations to perform before stopping.
Initialize: $S_{\mathbf{i}} \leftarrow \emptyset$.
Initialize: $h_{\mathbf{i}} \leftarrow h_0$.
Initialize: $\mathcal{L}_{\text{BEST}} \leftarrow \infty$.
Initialize: $\texttt{COUNTER} \leftarrow 0$.
Initialize: $(\overline{\boldsymbol{x}},\overline{y}) \leftarrow \text{argmax}_{(\boldsymbol{x},y)\in S} \ell^{\text{RMSE}}(h_0,\boldsymbol{x},y)$
while $\texttt{COUNTER} \leq T$ **do**
  $S_{\mathbf{i}} \leftarrow S_{\mathbf{i}} \cup \{(\overline{\boldsymbol{x}},\overline{y})\}$
  $h_{\mathbf{i}} \leftarrow A(S_{\mathbf{i}})$
  $(\overline{\boldsymbol{x}},\overline{y}) \leftarrow \text{argmax}_{(\boldsymbol{x},y)\in S_{\mathbf{i}^c}} \ell^{\text{RMSE}}(h_{\mathbf{i}},\boldsymbol{x},y)$
  if $\mathcal{L}^{RMSE}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}}) < \mathcal{L}_{BEST}$ **then**
    $\mathcal{L}_{\text{BEST}} \leftarrow \mathcal{L}^{\text{RMSE}}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}})$
    $\texttt{COUNTER} \leftarrow 0$
  else
    $\texttt{COUNTER} \leftarrow \texttt{COUNTER}+1$
  end
end
return $h_{\mathbf{i}}$

Table 12: Results for the decision forests trained using P2L. We report the RMSE achieved by the models with the smallest validation loss. The ratio $|\mathbf{i}|/n$ is presented in percents (%).

Dataset	Train loss	Validation loss	Test loss	kl bound	Linear bound	$\lvert\mathbf{i}\rvert/n$	Baseline test loss	$\ell^{\max}$
Powerplant	4.15±0.17	4.47±0.21	4.51±0.21	18.66±1.71	25.74±2.06	2.13±0.53	3.59±0.13	90.6
Infrared	0.21±0.01	0.23±0.02	0.26±0.01	1.19±0.06	1.48±0.06	3.60±0.46	0.23±0.01	4.26
Airfoil	1.78±0.21	2.28±0.29	2.37±0.17	21.91±1.03	25.41±1.00	14.61±2.03	2.10±0.15	45.13
Parkinson	3.48±0.44	3.69±0.36	3.83±0.45	17.23±0.55	19.24±0.85	7.13±1.22	2.23±0.16	45.13
Concrete	4.32±0.49	5.54±0.71	5.43±0.57	45.58±4.43	51.71±3.69	14.87±4.19	4.70±0.36	90.63

Table 13: Results for the decision trees trained using P2L. We report the RMSE achieved by the models with the smallest validation loss. The ratio $|\mathbf{i}|/n$ is presented in percents (%).

Dataset	Train loss	Validation loss	Test loss	kl bound	Linear bound	$\lvert\mathbf{i}\rvert/n$	Baseline test loss	$\ell^{\max}$
Powerplant	8.85±0.98	9.02±0.84	9.17±1.01	24.74±1.42	29.19±1.68	1.86±0.51	4.07±0.13	90.6
Infrared	0.24±0.02	0.27±0.03	0.31±0.04	1.76±0.05	2.08±0.04	8.39±0.48	0.27±0.03	4.26
Airfoil	6.54±0.89	6.55±0.54	6.41±0.44	28.94±0.71	30.73±0.65	15.70±2.83	3.01±0.19	45.13
Parkinson	12.19±1.82	12.20±1.73	12.09±2.01	21.91±0.48	22.18±0.55	2.47±1.31	3.20±0.15	41.37
Concrete	10.24±1.13	10.93±1.43	10.69±0.98	59.40±2.03	62.39±1.84	19.83±3.38	6.22±0.91	90.63

To compute the linear bound, we assume that the distribution is ς^2-sub-Gaussian with

$$\varsigma = \tfrac{1}{2}(y_+ - y_-).$$

We report the assumed lower and greater values in Table 14.

A.1.4 Amazon Polarity

We trained DistilBERT on the Amazon Reviews Polarity dataset. We use a subset of 10% of the real dataset, amounting to 360k datapoints. 180k are used for pretraining the model, 144k for training and 36k for validation. We use the given test set of 400k datapoints. Using P2L, we add 32 datapoints at a time to the compression set and stop the training when the validation loss hasn't decreased in 20 iterations. The initial model h_0 is the model pretrained on the 180k datapoints. For both P2L and the baseline, which was trained for 200 epochs or until it

Table 14: Minimum and maximum target values used to compute the bound on regression problems.

Dataset	Assumed lower target value	Observed minimum target value (in S)	Observed maximum target value (in S)	Assumed upper target value
Powerplant	412.71	420.26	495.76	503.31
Infrared	35.40	35.75	39.3	39.66
Airfoil	99.62	103.38	140.99	144.75
Parkinson	1.59	5.04	39.51	42.96
Concrete	0	2.33	82.6	90.63

reached 0 errors on the training dataset, we use the following hyperparameter grid :

- Number of pretraining epochs : $\{2,5\}$
- Pretraining learning rate : 2×10^{-5}
- Dropout probability : $\{0.1,0.2\}$
- Training learning rate : $\{10^{-6},10^{-7},10^{-8}\}$

B THEORETICAL RESULTS FROM THE LITERATURE

Corollary 10 (Laviolette et al. (2005), Corollary 1)**.** *For any distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, for any set of messages $\{M(\mathbf{i})\forall\mathbf{i}\in I\}$, for any deterministic reconstruction function $\mathcal{R}$ that outputs sample-compressed predictors $h\in\mathcal{H}$ and for any $\delta\in(0,1]$, with probability at least $1-\delta$ over the draw of $S\sim\mathcal{D}^n$, we have*

$$\forall\mathbf{i}\in I,\forall\sigma\in M(\mathbf{i})\colon R_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma))\leq 1-\exp\left(\frac{-1}{n-|\mathbf{i}|-\kappa}\left[\ln\binom{n-|\mathbf{i}|}{\kappa}+\ln\binom{n}{|\mathbf{i}|}+\ln\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right]\right),$$

with $\kappa=|\mathbf{i}^c|R_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma))$.

Theorem 11 (Pérez-Ortiz et al. (2021))**.** *For any distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, for any set $\mathcal{H}$ of predictors $h:\mathcal{X}\to\mathcal{Y}$, for any loss $\ell:\mathcal{H}\times\mathcal{X}\times\mathcal{Y}\to[0,1]$, for any dataset-independent prior distribution $\mathcal{P}$ on $\mathcal{H}$, for any $\delta,\delta'\in(0,1]$, with probability at least $1-\delta-\delta'$ over the draw of $S\sim\mathcal{D}^n$ and a set of m predictors $h_1,\ldots,h_m\sim\mathcal{Q}_S$, where $\mathcal{Q}_S$ is a dataset-dependent posterior distribution over $\mathcal{H}$, we have*

$$\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\mathcal{L}_{\mathcal{D}}(h)\leq \mathrm{kl}^{-1}\left(\mathrm{kl}^{-1}\left(\frac{1}{m}\sum_{i=1}^{m}\widehat{\mathcal{L}}_S(h_i),\frac{1}{m}\log\frac{2}{\delta'}\right),\frac{1}{n}\left[\mathrm{KL}(\mathcal{Q}||\mathcal{P})+\ln\left(\frac{2\sqrt{n}}{\delta}\right)\right]\right).$$

C PROOFS

C.1 Proof of the main result

Before proving Theorem 3, we restate Chernoff's bound in a way that will be useful to prove Theorem 3.

Lemma 12 (Chernoff's bound)**.** *For $t>0$ and X a random variable :*

$$\mathbb{P}\left(X\leq\frac{1}{t}\left[\ln\mathbb{E}e^{tX}+\ln\frac{1}{\delta}\right]\right)\geq 1-\delta.$$

Proof of Lemma 12. Chernoff's bound states that for a random variable X, any $t>0$ and $\epsilon>0$, we have

$$\mathbb{P}(X>\epsilon)\leq e^{-t\epsilon}\mathbb{E}e^{tX}.$$

By choosing $\delta = e^{-t\epsilon}\mathbb{E}e^{tX}$, we have

$$\begin{aligned}
&\delta = e^{-t\epsilon}\mathbb{E}e^{tX}\\
\iff &e^{t\epsilon} = \frac{1}{\delta}\mathbb{E}e^{tX}\\
\iff &t\epsilon = \ln\frac{1}{\delta}\mathbb{E}e^{tX}\\
\iff &\epsilon = \frac{1}{t}\left[\ln\mathbb{E}e^{tX} + \ln\frac{1}{\delta}\right].
\end{aligned}$$

Thus, we obtain

$$\mathbb{P}\left(X > \frac{1}{t}\left[\ln\mathbb{E}e^{tX} + \ln\frac{1}{\delta}\right]\right) \leq \delta.$$

□

Theorem 3. *For any distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, for any family of set of messages $\{M(\mathbf{i})|\mathbf{i}\in\mathcal{P}(n)\}$, for any deterministic reconstruction function $\mathcal{R}$ that outputs sample-compressed predictors $h\in\overline{\mathcal{H}}$, for any loss $\ell:\mathcal{H}\times\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$, for any comparator 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 $S\sim\mathcal{D}^n$, we have*

$$\forall\mathbf{i}\in\mathcal{P}(n),\sigma\in M(\mathbf{i}):\Delta\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma))\Big) \leq \frac{1}{|\mathbf{i}^c|}\left[\log\binom{n}{|\mathbf{i}|} + \log\left(\frac{\mathcal{E}_\Delta(\mathbf{i},\sigma)}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right],$$

with

$$\mathcal{E}_\Delta(\mathbf{i},\sigma) = \mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{|\mathbf{i}^c|}} e^{|\mathbf{i}^c|\Delta\left(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}.$$

Proof of Theorem 3. We start by defining the set of sample-compressed predictors. Given a dataset $S\sim\mathcal{D}^n$ and $\mathcal{H}$ a predictor set, we consider the following subset of $\mathcal{H}$, that contains only sample-compressed predictors :

$$\widehat{\mathcal{H}}_S \coloneqq \{\mathcal{R}(S_{\mathbf{i}},\sigma)|\mathbf{i}\in\mathcal{P}(n),\sigma\in M(\mathbf{i})\}\subseteq\mathcal{H}.$$

Note that for any vector of indices $\mathbf{i}\in\mathcal{P}(n)$ and any message $\sigma\in M(\mathbf{i})$, when given a dataset S, we obtain a predictor $\mathcal{R}(S_{\mathbf{i}},\sigma)\in\widehat{\mathcal{H}}_S$.

For a specific pair $(\mathbf{i},\sigma)$, let's study the value of $\Delta\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma))\Big)$, a realization of a random variable of mean

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}^n}\Delta\Big(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\Big) = \mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}\Delta\Big(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\Big).$$

With $\delta_{\mathbf{i}}^\sigma\in(0,1)$ and $t>0$, using Chernoff's bound as stated in Lemma 12, we have

$$\begin{aligned}
&\mathop{\mathbb{P}}_{S\sim\mathcal{D}^n}\left(\Delta\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}}^\sigma),\mathcal{L}_{\mathcal{D}}(h_{\mathbf{i}}^\sigma)\Big) \leq \frac{1}{t}\left[\ln\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}} e^{t\Delta\left(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)} + \ln\frac{1}{\delta_{\mathbf{i}}^\sigma}\right]\right)\\
&\geq 1-\delta_{\mathbf{i}}^\sigma.
\end{aligned}$$

Thanks to the union bound, we get a bound that is valid for all pairs $(\mathbf{i},\sigma)$ simultaneously,

$$\begin{aligned}
&\mathop{\mathbb{P}}_{S\sim\mathcal{D}^n}\left(\forall\mathbf{i}\in\mathcal{P}(n),\sigma\in M(\mathbf{i}):\Delta\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(h_{\mathbf{i}}^\sigma),\mathcal{L}_{\mathcal{D}}(h_{\mathbf{i}}^\sigma)\Big) \leq \frac{1}{t}\left[\ln\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}} e^{t\Delta\left(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)} + \ln\frac{1}{\delta_{\mathbf{i}}^\sigma}\right]\right)\\
&\geq 1-\sum_{\mathbf{i}\in\mathcal{P}(n)}\sum_{\sigma\in M(\mathbf{i})}\delta_{\mathbf{i}}^\sigma.
\end{aligned} \tag{7}$$

Given $\delta \in (0,1)$, we set $\delta_{\mathbf{i}}^{\sigma} = \binom{n}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|) P_{M(\mathbf{i})}(\sigma)\delta$ and we obtain

$$\begin{aligned}
\sum_{\mathbf{i}\in I}\sum_{\sigma\in M(\mathbf{i})} \delta_{\mathbf{i}}^{\sigma} &= \sum_{\mathbf{i}\in I}\sum_{\sigma\in M(\mathbf{i})} \binom{n}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|) P_{M(\mathbf{i})}(\sigma)\delta \\
&\leq \sum_{\mathbf{i}\in I} \binom{n}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|)\delta \\
&= \sum_{m=1}^{n}\sum_{\mathbf{i}\in I_m} \binom{n}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|)\delta \\
&= \sum_{m=1}^{n} \zeta(|\mathbf{i}|)\delta \\
&\leq \delta.
\end{aligned}$$

Thus, we have $1 - \sum_{\mathbf{i}\in I}\sum_{\sigma\in M(\mathbf{i})} \delta_{\mathbf{i}}^{\sigma} \geq 1-\delta$. We substitute $\delta_{\mathbf{i}}^{\sigma}$ by $\binom{n}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta$ and let $t = n - |\mathbf{i}|$ in Equation (7) to finish the proof. □

Note that the proof of Theorem 3 relies on specific choices for the values of the variables $\delta_{\mathbf{i}}^{\sigma}$ and t in Equation (7). The next paragraphs discuss the rationales behind these choices.

The choice of $t = n - |\mathbf{i}|$. To turn Equation (7) into a computable bound, one needs to either compute or upper bound the following term:

$$\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}} \mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}} e^{t\Delta\left(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}.$$

As the set T is a realization of a $n-|\mathbf{i}|$ datapoints, choosing $t = n-|\mathbf{i}|$ generally ensure that this term is bounded. Indeed, this is a requirement for the proofs of multiple results in the PAC-Bayesian theory.

The choice of $\delta_{\mathbf{i}}^{\sigma} = \binom{n}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta$. Importantly, the value of $\delta_{\mathbf{i}}^{\sigma}$ needs to be defined independently of S. Thus, choosing $\delta_{\mathbf{i}}^{\sigma}$ is equivalent to choosing the prior distributions $P_{\mathcal{P}(n)}$ and $P_{M(\mathbf{i})}$ in order to obtain

$$\delta_{\mathbf{i}}^{\sigma} = P_{\mathcal{P}(n)}(\mathbf{i}) P_{M(\mathbf{i})}(\sigma)\delta. \tag{8}$$

Consider the set $I_m = \{\mathbf{i} \in \mathcal{P}(n) : |\mathbf{i}| = m\}$. As we have no information on which $\mathbf{i} \in I_m$ is likely to lead to a good compression set for the reconstruction $\mathcal{R}$, we define a uniform distribution over all $\binom{n}{m}$ vectors in I_m, which gives a weight of $\binom{n}{m}^{-1} \forall \mathbf{i} \in I_m$. Now, we want consider all possible sizes of compression set

$$I = \bigcup_{k=0}^{n} I_k,$$

so we need to define a probability distribution over each set I_k. We could simply choose $\frac{1}{n+1}$, but the probabilities would tend very fast to zero when we consider a large number n of compression set sizes. It is a better choice, as discussed by Marchand and Sokolova (2005) in Section 5.2, to choose :

$$\zeta(m) = \frac{6}{\pi^2(m+1)^2}, \quad \text{for which } \sum_{m=0}^{\infty}\zeta(m) = 1.$$

Then, Equation (8) is applied with $P_{\mathcal{P}(n)}(\mathbf{i}) = \binom{n}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|)$. To apply the theorem, one must also provide choice of prior distributions $P_{M(\mathbf{i})}$ over the messages $M(\mathbf{i})$ such that $\sum_{\sigma\in M(\mathbf{i})} P_{M(\mathbf{i})}(\sigma) \leq 1$ for all $\mathbf{i} \in \mathcal{P}(n)$.

C.2 Corollaries to the main result

To prove most corollaries, we are going to need the following lemma.

Lemma 13 (Maurer (2004), Germain et al. (2015))**.** *Let X be any random variable with values in $[0,1]$ and expectation $\mu=\mathbb{E}(X)$. Denote X the vector containing the results of n independent realizations of X. Then, consider a Bernoulli random variable X' ($\{0,1\}$-valued) of probability of success μ. Denote $X'\in\{0,1\}^n$ the vector containing the results of n independent realizations of X'.*

If function $g:[0,1]^n\to\mathbb{R}$ is convex, then

$$\mathbb{E}[g(X)] \leq \mathbb{E}[g(X')].$$

We now prove our first corollary.

Corollary 4. *In the setting of Theorem 3, for any $C>0$, for any loss function $\ell:\mathcal{H}\times\mathcal{X}\times\mathcal{Y}\to[0,1]$, with probability at least $1-\delta$ over the draw of $S\sim\mathcal{D}^n$, we have*

$$\forall \mathbf{i}\in\mathcal{P}(n),\sigma\in M(\mathbf{i}):\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma))\leq\frac{1-\exp(-\epsilon_C(\mathbf{i},\sigma,\delta))}{1-e^{-C}},$$

with

$$\epsilon_C(\mathbf{i},\sigma,\delta)=C\widehat{\mathcal{L}}_{S_{\bar{\mathbf{i}}}}(\mathcal{R}(S_{\mathbf{i}},\sigma))+\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right].$$

Proof of Corollary 4. The proof is divided in two steps: bounding $\mathcal{E}_{\Delta_C}$ and rearranging the terms. Note that these both steps are common in the proofs of PAC-Bayesian literature. Our proof mainly follow the one of Germain et al. (2015).

We start by bounding $\mathcal{E}_{\Delta_C}$. Let us introduce a random variable $X_{\mathbf{i}}^{\sigma}$ that follows a binomial distribution of m trials with a probability of success $\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))$, denoted $B(m,\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))$. We use Lemma 13 with $g(\cdot)=e^{m\Delta_C(\cdot,\mathcal{L}_{\mathcal{D}}(h))}$.

$$\begin{aligned}
\mathcal{E}_{\Delta_C}(\mathbf{i},\sigma)&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\bar{\mathbf{i}}}\sim\mathcal{D}^{n-|\mathbf{i}|}}e^{(n-|\mathbf{i}|)\Delta_C\left(\widehat{\mathcal{L}}_{T_{\bar{\mathbf{i}}}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&\leq\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{X_{\mathbf{i}}^{\sigma}\sim B(m,\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))}e^{(n-|\mathbf{i}|)\Delta_C\left(\frac{1}{n-|\mathbf{i}|}X_{\mathbf{i}}^{\sigma},\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\sum_{k=0}^{n-|\mathbf{i}|}\mathop{\mathbb{P}}_{X_{\mathbf{i}}^{\sigma}\sim B(m,\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))}(X_{\mathbf{i}}^{\sigma}=k)e^{(n-|\mathbf{i}|)\Delta_C\left(\frac{k}{n-|\mathbf{i}|},\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))^k(1-\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))^{n-|\mathbf{i}|-k}e^{(n-|\mathbf{i}|)\Delta_C\left(\frac{k}{n-|\mathbf{i}|},\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&\leq\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^k(1-r)^{n-|\mathbf{i}|-k}e^{(n-|\mathbf{i}|)\Delta_C\left(\frac{k}{n-|\mathbf{i}|},r\right)}\right]\\
&=\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^k(1-r)^{n-|\mathbf{i}|-k}e^{(n-|\mathbf{i}|)\Delta_C\left(\frac{k}{n-|\mathbf{i}|},r\right)}\right]\\
&=\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^k(1-r)^{n-|\mathbf{i}|-k}\frac{e^{-Ck}}{[1-(1-e^{-C})r]^{n-|\mathbf{i}|}}\right]\\
&=\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}\left(re^{-C}\right)^k(1-r)^{n-|\mathbf{i}|-k}\frac{1}{[1-(1-e^{-C})r]^{n-|\mathbf{i}|}}\right]\\
&=\sup_{r\in[0,1]}\left[\frac{[re^{-C}+(1-r)]^{n-|\mathbf{i}|}}{[1-(1-e^{-C})r]^{n-|\mathbf{i}|}}\right]=\sup_{r\in[0,1]}[1]=1.
\end{aligned}$$

where the last line is derived using binomial theorem.

Now, we rearrange the terms:

$$\Delta\left(\widehat{\mathcal{L}}_{S_{\bar{\mathbf{i}}}}(R(\mathbf{i},\sigma)),\mathcal{L}_D(R(\mathbf{i},\sigma))\right)\leq\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right]$$

$$-\ln\big(1-\mathcal{L}_D(R(\mathbf{i},\sigma))(1-e^{-C})\big)-C\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(R(\mathbf{i},\sigma))\leq\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right]$$

$$\ln\big(1-\mathcal{L}_D(R(\mathbf{i},\sigma))(1-e^{-C})\big)\geq-C\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(R(\mathbf{i},\sigma))-\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right]$$

$$1-\mathcal{L}_D(R(\mathbf{i},\sigma))(1-e^{-C})\geq\exp\left(-C\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(R(\mathbf{i},\sigma))-\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right]\right)$$

$$\mathcal{L}_D(R(\mathbf{i},\sigma))(1-e^{-C})\leq1-\exp\left(-C\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(R(\mathbf{i},\sigma))-\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right]\right)$$

$$\mathcal{L}_D(R(\mathbf{i},\sigma))\leq\frac{1}{1-e^{-C}}\left[1-\exp\left(-C\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(R(\mathbf{i},\sigma))-\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right]\right)\right]$$

□

Corollary 6. *In the setting of Theorem 3, for any loss function $\ell:\mathcal{H}\times\mathcal{X}\times\mathcal{Y}\to[0,1]$, with probability at least $1-\delta$ over the draw of $S\sim\mathcal{D}^n$, we have*

$$\forall\mathbf{i}\in\mathcal{P}(n),\sigma\in M(\mathbf{i}):\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma))\leq\mathrm{kl}^{-1}\Big(\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma)),\epsilon_{\mathrm{kl}}(\mathbf{i},\sigma,\delta)\Big),$$

with $\mathrm{kl}^{-1}(q,\epsilon)=\mathrm{argsup}_{0\leq p\leq1}\{\mathrm{kl}(q,p)\leq\epsilon\}$ *and*

$$\epsilon_{\mathrm{kl}}(\mathbf{i},\sigma,\delta)=\frac{1}{n-|\mathbf{i}|}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{2\sqrt{n-|\mathbf{i}|}}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right].$$

Proof. To prove this corollary, we need to bound $\mathcal{E}_{\mathrm{kl}}$. In the PAC-Bayes literature, this was first done by Langford and Seeger (2001); Seeger (2002) and then improved by Maurer (2004). We restate the proof of the latter for completeness.

We use Lemma 13 with $g(\cdot)=e^{m\mathrm{kl}(\cdot,\mathcal{L}_{\mathcal{D}}(h))}$.

$$\begin{aligned}
\mathcal{E}_{\mathrm{kl}}(\mathbf{i},\sigma)&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}e^{(n-|\mathbf{i}|)\mathrm{kl}\left(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&\leq\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{X_{\mathbf{i}}^{\sigma}\sim B(m,\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))}e^{(n-|\mathbf{i}|)\mathrm{kl}\left(\frac{1}{n-|\mathbf{i}|}X_{\mathbf{i}}^{\sigma},\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\sum_{k=0}^{n-|\mathbf{i}|}\mathop{\mathbb{P}}_{X_{\mathbf{i}}^{\sigma}\sim B(m,\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))}(X_{\mathbf{i}}^{\sigma}=k)e^{(n-|\mathbf{i}|)\mathrm{kl}\left(\frac{k}{n-|\mathbf{i}|},\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))^{k}(1-\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma)))^{n-|\mathbf{i}|-k}e^{(n-|\mathbf{i}|)\mathrm{kl}\left(\frac{k}{n-|\mathbf{i}|},\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\right)}\\
&\leq\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^{k}(1-r)^{n-|\mathbf{i}|-k}e^{(n-|\mathbf{i}|)\mathrm{kl}\left(\frac{k}{n-|\mathbf{i}|},r\right)}\right]\\
&=\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^{k}(1-r)^{n-|\mathbf{i}|-k}e^{(n-|\mathbf{i}|)\mathrm{kl}\left(\frac{k}{n-|\mathbf{i}|},r\right)}\right]\\
&=\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^{k}(1-r)^{n-|\mathbf{i}|-k}\times e^{(n-|\mathbf{i}|)\left(\frac{k}{n-|\mathbf{i}|}\ln\left(\frac{k}{n-|\mathbf{i}|}\cdot\frac{1}{r}\right)+\left(1-\frac{k}{n-|\mathbf{i}|}\right)\ln\left(\left(1-\frac{k}{n-|\mathbf{i}|}\right)\cdot\frac{1}{1-r}\right)\right)}\right]\\
&=\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^{k}(1-r)^{n-|\mathbf{i}|-k}\times e^{k\ln\left(\frac{k}{n-|\mathbf{i}|}\cdot\frac{1}{r}\right)+(n-|\mathbf{i}|-k)\ln\left(\left(1-\frac{k}{n-|\mathbf{i}|}\right)\cdot\frac{1}{1-r}\right)}\right]\\
&=\sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^{k}(1-r)^{n-|\mathbf{i}|-k}\times e^{\ln\left(\frac{k}{n-|\mathbf{i}|}\right)^{k}+\ln\left(\frac{1}{r}\right)^{k}+\ln\left(1-\frac{k}{n-|\mathbf{i}|}\right)^{n-|\mathbf{i}|-k}+\ln\left(\frac{1}{1-r}\right)^{n-|\mathbf{i}|-k}}\right]
\end{aligned}$$

$$
\begin{aligned}
&= \sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}(r)^k(1-r)^{n-|\mathbf{i}|-k}\times\frac{1}{(r)^k(1-r)^{n-|\mathbf{i}|-k}}\left(\frac{k}{n-|\mathbf{i}|}\right)^k\left(1-\frac{k}{n-|\mathbf{i}|}\right)^{n-|\mathbf{i}|-k}\right]\\
&= \sup_{r\in[0,1]}\left[\sum_{k=0}^{n-|\mathbf{i}|}\binom{n-|\mathbf{i}|}{k}\left(\frac{k}{n-|\mathbf{i}|}\right)^k\left(1-\frac{k}{n-|\mathbf{i}|}\right)^{n-|\mathbf{i}|-k}\right]\\
&\leq e^{\frac{1}{12(n-|\mathbf{i}|)}}\sqrt{\frac{\pi(n-|\mathbf{i}|)}{2}}+2\\
&\leq 2\sqrt{n-|\mathbf{i}|}.
\end{aligned}
$$

The last two inequalities were proven by Maurer (2004) for $n-|\mathbf{i}|\geq 8$. As noticed afterward by Germain et al. (2015), it can be verified numerically that $\mathcal{E}_{\mathrm{kl}}(\mathbf{i},\sigma)\leq 2\sqrt{n-|\mathbf{i}|}$ also holds for $1\leq n-|\mathbf{i}|<8$.. □

Corollary 7. *In the setting of Theorem 3, for any $\lambda>0$, with a ς^2-sub-Gaussian loss function $\ell:\mathcal{H}\times\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$, with probability at least $1-\delta$ over the draw of $S\sim\mathcal{D}^n$, we have*

$$
\forall\mathbf{i}\in\mathcal{P}(n),\sigma\in M(\mathbf{i}):\mathcal{L}_{\mathcal{D}}(\mathcal{R}(S_{\mathbf{i}},\sigma))\leq\widehat{\mathcal{L}}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma))+\frac{\lambda\varsigma^2}{2}+\frac{1}{\lambda(n-|\mathbf{i}|)}\left[\log\binom{n}{|\mathbf{i}|}+\log\left(\frac{1}{\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta}\right)\right].
$$

Proof. We assume that the loss ℓ is ς^2-sub-Gaussian, which is defined as

$$
\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\exp\left[\lambda\left(\ell(h,\boldsymbol{x},y)-\mathop{\mathbb{E}}_{(\boldsymbol{x}',y')\sim\mathcal{D}}\ell(h,\boldsymbol{x}',y')\right)\right]\leq\exp\left(\frac{\lambda^2\varsigma^2}{2}\right).
$$

Then, we have

$$
\begin{aligned}
\mathcal{E}_{\Delta_\lambda}(\mathbf{i},\sigma)&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}\exp\Big[(n-|\mathbf{i}|)\Delta_\lambda\Big(\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma)),\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\Big)\Big]\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}\exp\Big[(n-|\mathbf{i}|)\lambda\Big(\mathcal{L}_{\mathcal{D}}(\mathcal{R}(T_{\mathbf{i}},\sigma))-\widehat{\mathcal{L}}_{T_{\mathbf{i}^c}}(\mathcal{R}(T_{\mathbf{i}},\sigma))\Big)\Big]\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}\exp\left[(n-|\mathbf{i}|)\lambda\left(\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x},y)-\frac{1}{n-|\mathbf{i}|}\sum_{i=1}^{n-|\mathbf{i}|}\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x}_i,y_i)\right)\right]\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}\exp\left[\lambda\sum_{i=1}^{n-|\mathbf{i}|}\left(\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x},y)-\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x}_i,y_i)\right)\right]\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}\exp\left[-\lambda\sum_{i=1}^{n-|\mathbf{i}|}\left(\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x}_i,y_i)-\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x},y)\right)\right]\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\mathop{\mathbb{E}}_{T_{\mathbf{i}^c}\sim\mathcal{D}^{n-|\mathbf{i}|}}\prod_{i=1}^{n-|\mathbf{i}|}\exp\left[-\lambda\left(\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x}_i,y_i)-\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x},y)\right)\right]\\
&=\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\prod_{i=1}^{n-|\mathbf{i}|}\mathop{\mathbb{E}}_{(\boldsymbol{x}_i,y_i)\sim\mathcal{D}}\exp\left[-\lambda\left(\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x}_i,y_i)-\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}}\ell(\mathcal{R}(T_{\mathbf{i}},\sigma),\boldsymbol{x},y)\right)\right] \qquad (9)\\
&\leq\mathop{\mathbb{E}}_{T_{\mathbf{i}}\sim\mathcal{D}^{|\mathbf{i}|}}\prod_{i=1}^{n-|\mathbf{i}|}\exp\left(\frac{\lambda^2\varsigma^2}{2}\right) \qquad (10)\\
&=\exp\left(\frac{(n-|\mathbf{i}|)\lambda^2\varsigma^2}{2}\right).
\end{aligned}
$$

Equation (9) relies on the *i.i.d.* assumption and Equation (10), on the ς^2-sub-Gaussian assumption.

We replace the comparator function in Theorem 3 and bound the cumulant generating function $\mathcal{E}_{\Delta_\lambda}$ to finish the proof. □

C.3 Behavior with zero error

Let us recall the definition of the binomial tail,

$$\mathrm{Bin}(k,m,r) = \sum_{i=0}^{k} \binom{m}{i} r^i (1-r)^{m-i},$$

and the binomial tail inversion,

$$\overline{\mathrm{Bin}}(k,m,\delta) = \operatorname*{argsup}_{r\in[0,1]} \{\mathrm{Bin}(k,m,r) \geq \delta\}.$$

The proofs of Theorem 8 and Theorem 9 make use of the following lemma.

Lemma 14. *In the consistent case, the binomial tail has the following analytical solution:*

$$\overline{\mathrm{Bin}}(0,m,\delta) = 1 - \exp\left(-\frac{1}{m}\ln\left(\frac{1}{\delta}\right)\right).$$

Proof. We start by rewriting the binomial tail distribution, with $k=0$.

$$\begin{aligned}
\mathrm{Bin}(0,m,r) &= \sum_{i=0}^{0} \binom{m}{i} r^i (1-r)^{m-i} \\
&= \binom{m}{0} r^0 (1-r)^{m-0} \\
&= 1 \cdot 1 \cdot (1-r)^m \\
&= (1-r)^m.
\end{aligned}$$

We now rewrite the binomial tail inversion.

$$\begin{aligned}
\overline{\mathrm{Bin}}(0,m,\delta) &= \operatorname*{argsup}_{r\in[0,1]} \{\mathrm{Bin}(0,m,r) \geq \delta\} \\
&= \operatorname*{argsup}_{r\in[0,1]} \{(1-r)^m \geq \delta\} \\
&= \operatorname*{argsup}_{r\in[0,1]} \left\{1 - r \geq \delta^{\frac{1}{m}}\right\} \\
&= \operatorname*{argsup}_{r\in[0,1]} \left\{1 - \delta^{\frac{1}{m}} \geq r\right\} \\
&= 1 - \delta^{\frac{1}{m}} \qquad (11) \\
&= 1 - \exp\left(\ln\left(\delta^{\frac{1}{m}}\right)\right) \\
&= 1 - \exp\left(-\frac{1}{m}\ln\left(\frac{1}{\delta}\right)\right)
\end{aligned}$$

At Eq. (11), we use the fact that the maximum value of r such that $1-\delta^{\frac{1}{m}} \geq r$ is simply $1-\delta^{\frac{1}{m}} = r$. □

We did not find any mention of Lemma 14's equality in the literature. However, the following inequality is well known and can be found in papers such as Laviolette et al. (2005):

$$\overline{\mathrm{Bin}}(k,m,\delta) \leq 1 - \exp\left(\frac{-1}{m-k}\left[\ln\binom{m}{k} + \ln\left(\frac{1}{\delta}\right)\right]\right).$$

When $k=0$, it reduces to

$$\overline{\mathrm{Bin}}(0,m,\delta) \leq 1 - \exp\left(\frac{-1}{m}\ln\left(\frac{1}{\delta}\right)\right).$$

We now prove Theorem 8.

Theorem 8. *In the consistent case, i.e. when $\widehat{R}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}},\sigma))=0$, Corollary 4 is arbitrarily close to the binomial tail inversion of Theorem 1. Indeed, we have*

$$\overline{\mathrm{Bin}}(0,|\mathbf{i}^c|,\delta_{\mathbf{i}}^{\sigma}) = \inf_{C>0}\left\{\frac{1-\exp\left(-\frac{1}{|\mathbf{i}^c|}\ln\frac{1}{\delta_{\mathbf{i}}^{\sigma}}\right)}{1-e^{-C}}\right\} \tag{3}$$

$$= \lim_{C\to\infty}\left\{\frac{1-\exp\left(-\frac{1}{|\mathbf{i}^c|}\ln\frac{1}{\delta_{\mathbf{i}}^{\sigma}}\right)}{1-e^{-C}}\right\} \tag{4}$$

with $\delta_{\mathbf{i}}^{\sigma}=\binom{n}{|\mathbf{i}|}^{-1}\zeta(|\mathbf{i}|)P_{M(\mathbf{i})}(\sigma)\delta$.

Proof. From Lemma 14, we have

$$\overline{\mathrm{Bin}}(0,m,\delta)=1-\exp\left(\frac{-1}{m}\ln\left(\frac{1}{\delta}\right)\right).$$

Thus, we will prove the two following equations :

$$1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)=\lim_{C\to\infty}\left\{\frac{1}{1-e^{-C}}\left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\right\} \tag{12}$$

and

$$\lim_{C\to\infty}\left\{\frac{1}{1-e^{-C}}\left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\right\}=\inf_{C>0}\left\{\frac{1}{1-e^{-C}}\left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\right\} \tag{13}$$

To prove Eq. (12), we simply compute

$$\begin{aligned}\lim_{C\to\infty}\frac{1}{1-e^{-C}}\left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right] &= \left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\lim_{C\to\infty}\frac{1}{1-e^{-C}}\\ &= \left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\cdot 1\\ &= 1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\end{aligned}$$

Let's now prove Eq. (13). We know that

$$\inf_{C>0}\left\{\frac{1}{1-e^{-C}}\left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\right\}=\left[1-\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\inf_{C>0}\left\{\frac{1}{1-e^{-C}}\right\}$$

We obtain that function $f(C)=\frac{1}{1-e^{-C}}$ is decreasing on $[0,\infty)$, and thus has a minimum at $C\to\infty$, by the fact that its derivative is negative for all $C\geq 0$:

$$\begin{aligned}f'(C)&=\frac{\mathrm{d}}{\mathrm{d}C}\frac{1}{1-e^{-C}}=\frac{\mathrm{d}}{\mathrm{d}C}\left(1-e^{-C}\right)^{-1}\\ &=\frac{-1}{\left(1-e^{-C}\right)^2}\frac{\mathrm{d}}{\mathrm{d}C}\left(1-e^{-C}\right)\\ &=\frac{-1}{\left(1-e^{-C}\right)^2}e^{-C}\\ &=(-1)\frac{e^{-C}}{\left(1-e^{-C}\right)^2}.\end{aligned}$$

□

Note that if one tries to find a value of $C\in\mathbb{R}_{>0}$ such that the derivative $f'(C)$ of the above proof is zero, they obtain

$$\begin{aligned}f'(C)=(-1)\frac{e^{-C}}{\left(1-e^{-C}\right)^2}&=0\\ \iff e^{-C}&=0.\end{aligned}$$

There is no $C \in \mathbb{R}_{>0}$ such that $e^{-C} = 0$, however we know that $\lim_{C\to\infty} e^{-C} = 0$.

Thus, as the function is monotonically decreasing when C increases, given an arbitrarily small $\epsilon > 0$, there is always a $C(\epsilon)$ large enough such that

$$\frac{1}{1-e^{-C(\epsilon)}} - 1 \leq \epsilon.$$

For example, with $\epsilon = 0.01$ and $C(\epsilon) = 4.616$, we have $\frac{1}{1-e^{-C(\epsilon)}} - 1 = 0.00999 \leq 0.01$. With $\epsilon = 10^{-5}$ and $C(\epsilon) = 11.513$, we have $\frac{1}{1-e^{-C(\epsilon)}} - 1 = 0.999 \times 10^{-5} \leq 10^{-5}$. Thus, it is possible to be arbitrarily tight to 1, for any $\epsilon > 0$.

We now prove Theorem 9.

Theorem 9. *In the consistent case, i.e. when $\widehat{R}_{S_{\mathbf{i}^c}}(\mathcal{R}(S_{\mathbf{i}}, \sigma)) = 0$, Corollary 6 is a tight upper bound of Theorem 1 up to a constant $K(m, \delta)$. Indeed, we have*

$$\overline{\mathrm{Bin}}(0, |\mathbf{i}^c|, \delta^\sigma_{\mathbf{i}}) \leq \mathrm{kl}^{-1}\left(0, \frac{1}{|\mathbf{i}^c|} \ln \frac{2\sqrt{|\mathbf{i}^c|}}{\delta^\sigma_{\mathbf{i}}}\right) \tag{5}$$

$$= \overline{\mathrm{Bin}}(0, |\mathbf{i}^c|, \delta^\sigma_{\mathbf{i}}) + K(|\mathbf{i}^c|, \delta^\sigma_{\mathbf{i}}), \tag{6}$$

with $K(m, \delta) = \exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right) - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right)$ *and* $\delta^\sigma_{\mathbf{i}} = \binom{n}{|\mathbf{i}|}^{-1} \zeta(|\mathbf{i}|) P_{M(\mathbf{i})}(\sigma)\delta$.

Proof. To prove this result, we need to prove the following sequence of equations:

$$\overline{\mathrm{Bin}}(0, m, \delta) = \mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{1}{\delta}\right) \tag{14}$$

$$\leq \mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right) \tag{15}$$

$$= \mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{1}{\delta}\right) + K(m, \delta). \tag{16}$$

We first prove Eq. (14). We already know that

$$\overline{\mathrm{Bin}}(0, m, \delta) = \inf_{C>0}\left\{\frac{1}{1-e^{-C}}\left[1 - \exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right]\right\}.$$

Moreover, from the proof of Theorem 3 of Letarte et al. (2019), for any constant $A > 0$, we know that

$$\inf_{C>0}\left\{\frac{1}{1-e^{-C}}\left[1 - \exp\left(-C\left[\frac{k}{m}\right] - \frac{1}{m}\ln\frac{A}{\delta}\right)\right]\right\} = \mathrm{kl}^{-1}\left(\frac{k}{m}, \frac{1}{m}\ln\frac{A}{\delta}\right). \tag{17}$$

Thus, with $A = 1$, we have

$$\overline{\mathrm{Bin}}(0, m, \delta) = \mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{1}{\delta}\right).$$

We now prove Eq. (15).

The function $\mathrm{kl}(0, p)$ is monotonically increasing and $\mathrm{kl}(0, 1) = \infty$. Thus, there exists a value $p^* = \mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{1}{\delta}\right)$ such that $\mathrm{kl}(0, p^*) = \frac{1}{m}\ln\frac{1}{\delta}$. Moreover, there exists a value $p^\dagger = \mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right)$ such that $\mathrm{kl}(0, p^\dagger) = \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}$. As $\mathrm{kl}(0, p)$ is monotonically increasing and

$$\mathrm{kl}(0, p^*) = \frac{1}{m}\ln\frac{1}{\delta} \leq \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta} = \mathrm{kl}(0, p^\dagger),$$

then $p^* \leq p^\dagger$.

Finally, we prove Eq. (16) and show that $K(m,\delta)$ tends to 0 when m tends to ∞ and is bounded by

$$0 \leq K(m,\delta) \leq \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}.$$

We start by defining the constant $K(m,\delta)$ as the gap between the two following terms:

$$\begin{aligned}
&\mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right) - \mathrm{kl}^{-1}\left(0, \frac{1}{m}\ln\frac{1}{\delta}\right) \\
&= \left[1 - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right)\right] - \left[1 - \exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right] \\
&= \exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right) - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right) \\
&=: K(m,\delta)
\end{aligned}$$

In the second line, we use Eq. (17) with both $A = 2\sqrt{m}$ and $A = 1$. □

We now highlight some properties of $K(m,\delta)$. First of all, we show that $K(m,\delta) \geq 0$ for $m \geq \frac{1}{4}$.

$$\begin{aligned}
&\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right) - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right) \geq 0 \\
\iff &\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right) \geq \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right) \\
\iff &-\frac{1}{m}\ln\frac{1}{\delta} \geq -\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta} \qquad (18) \\
\iff &\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta} \geq \frac{1}{m}\ln\frac{1}{\delta} \\
\iff &2\sqrt{m} \geq 1 \qquad (19) \\
\iff &m \geq \frac{1}{4}
\end{aligned}$$

In Eq. (18) and Eq. (19), we use the fact that both the exponentials and logarithms are increasing functions. From this, we also know that $K(\frac{1}{4},\delta) = 0$. As the parameter m is the size of a dataset, we know that we always have $m \geq 1$ and thus $K(m,\delta) \geq 0$.

Secondly, we show that $K(m,\delta)$ tends to 0 when m tends to ∞.

$$\begin{aligned}
\lim_{m\to\infty} K(m,\delta) &= \lim_{m\to\infty}\left\{\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right) - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right)\right\} \\
&= \lim_{m\to\infty}\left\{\exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right)\right\} - \lim_{m\to\infty}\left\{\exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right)\right\} \\
&= 1 - 1 = 0.
\end{aligned}$$

Next, we compute a simple upper bound of $K(m,\delta)$, that also tends to 0 when $m \to \infty$.

$$\begin{aligned}
K(m,\delta) &= \exp\left(-\frac{1}{m}\ln\frac{1}{\delta}\right) - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right) \\
&\leq 1 - \exp\left(-\frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}\right) \\
&\leq \frac{1}{m}\ln\frac{2\sqrt{m}}{\delta}.
\end{aligned}$$

The last line uses the inequality $1 - e^{-x} \leq x$.