PAC-Bayesian Contrastive Unsupervised Representation Learning

Kento Nozawa
The University of Tokyo & RIKEN
Japan

Pascal Germain
Université Laval
Canada

Benjamin Guedj
Inria & University College London
France and United Kingdom

Abstract

Contrastive unsupervised representation learning (CURL) is the state-of-the-art technique to learn representations (as a set of features) from unlabelled data. While CURL has collected several empirical successes recently, theoretical understanding of its performance was still missing. In a recent work, Arora et al. (2019) provide the first generalisation bounds for CURL, relying on a Rademacher complexity. We extend their framework to the flexible PAC-Bayes setting, allowing us to deal with the non-iid setting. We present PAC-Bayesian generalisation bounds for CURL, which are then used to derive a new representation learning algorithm. Numerical experiments on real-life datasets illustrate that our algorithm achieves competitive accuracy, and yields non-vacuous generalisation bounds.

1 INTRODUCTION

Unsupervised representation learning (Bengio et al., 2013) aims at extracting features representation from an unlabelled dataset for downstream tasks such as classification and clustering (see Mikolov et al., 2013; Noroozi and Favaro, 2016; Zhang et al., 2016; Caron et al., 2018; Devlin et al., 2019). An unsupervised representation learning model is typically learnt by solving a pretext task without supervised information. Trained model work as a feature extractor for supervised tasks.

In unsupervised representation learning, contrastive loss is a widely used objective function class. Contrastive loss uses two types of data pair, namely, similar pair and dissimilar pair. Their similarity is defined without label information of a supervised task. For example, in word representation learning, Mikolov et al. (2013) define a similar pair as co-occurrence words in the same context, while dissimilar pairs are randomly sampled from a fixed distribution. Intuitively, by minimising a contrastive loss, similar data samples are mapped to similar representations in feature space in terms of some underlying metric (as the inner product), and dissimilar samples are not mapped to similar representations.

Contrastive unsupervised representation learning improves the performance of supervised models in practice, and has attracted a lot of research interest lately (see Chen et al., 2020, and references therein), although usage is still quite far ahead of theoretical understanding. Recently, Arora et al. (2019) introduced a theoretical framework for contrastive unsupervised representation learning and derived the first generalisation bounds for CURL. In parallel, PAC-Bayes is emerging as a principled tool to understand and quantify the generalisation ability of many machine learning algorithms, including deep neural networks (as recently studied by Dziugaite and Roy, 2017; Neyshabur et al., 2018; Letarte et al., 2019).

Our contributions. We extend the framework introduced by Arora et al. (2019), by adopting a PAC-Bayes approach to contrastive unsupervised representation learning. We derive the first PAC-Bayes generalisation bounds for CURL, both in iid and non-iid settings. Our bounds are then used to derive new CURL algorithms, for which we provide a complete implementation. The paper closes with numerical experiments on two real-life datasets (`CIFAR-100` and `AUSLAN`) showing that our bounds are non-vacuous in the iid setting.

2 CONTRASTIVE UNSUPERVISED REPRESENTATION LEARNING

2.1 LEARNING FRAMEWORK

Inputs are denoted $\mathbf{x} \in X = \mathbb{R}^{d_0}$, and outputs are denoted $y \in Y$, where Y is a discrete and finite set.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

The *representation* is learnt from a (large) unlabelled dataset $U = \{\mathbf{z}_i\}_{i=1}^m$, where $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_{i1}^-, \ldots, \mathbf{x}_{ik}^-)$ is a tuple of $k{+}2$ elements; $\mathbf{x}_i$ being *similar* to $\mathbf{x}_i^+$ and *dissimilar* to every element of the *negative sample set* $\{\mathbf{x}_{ij}^-\}_{j=1}^k$. The predictor is learnt from a labelled dataset $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$.

In the following, we present the contrastive framework proposed by Arora et al. (2019) in a simplified scenario in order to highlight the key ideas, where the supervised prediction task is binary and the negative sample sets for unsupervised representation learning contain one element. Thus, we choose the label set to be $Y = \{-1, 1\}$, and the unsupervised set U contains triplets $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)$. The extension to a more generic setting (for $|Y| > 2$ and $k > 1$) bears no particular difficulty and is deferred to Appendix A.2. It is important to note at this stage that both U and S are assumed to be **iid** (independent, identically distributed) collections, as also assumed by Arora et al. (2019).

Latent classes and data distributions. The main assumption is the existence of a set of *latent classes* $\mathcal{C}$. Let us denote by ρ a probability distribution over $\mathcal{C}$. Moreover, with each class $c \in \mathcal{C}$, comes a class distribution $\mathcal{D}_c$ over the input space X. A similar pair $(\mathbf{x}, \mathbf{x}^+)$ is such that both $\mathbf{x}$ and $\mathbf{x}^+$ are generated by the same class distribution. Note that an input $\mathbf{x}$ possibly belongs to multiple classes: take the example of $\mathbf{x}$ being an image and $\mathcal{C}$ a set of latent classes including "the image depicts a dog" and "the image depicts a cat" (both classes are not mutually exclusive).

Definition 1. Let ρ^2 be a shorthand for the joint distribution (ρ, ρ). We refer to the *unsupervised data distribution* $\mathcal{U}$ as the process that generates an unlabelled sample $\mathbf{z} = (\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-)$ according to the following scheme:
1. Draw two latent classes $(c^+, c^-) \sim \rho^2$;
2. Draw two similar samples $(\mathbf{x}, \mathbf{x}^+) \sim (\mathcal{D}_{c^+})^2$;
3. Draw a negative sample $\mathbf{x}^- \sim \mathcal{D}_{c^-}$.

The labelled sample S is obtained by fixing two classes $c^\pm = \{c^-, c^+\} \in \mathcal{C}^2$ (from now on, the shorthand notation $c^\pm$ is used to refer to a pair of latent classes). Each class is then mapped on a label of Y. We fix $y_{c^-} = -1$ and $y_{c^+} = 1$; Thus we can write $Y = \{y_{c^-}, y_{c^+}\}$ as an ordered set. The label is obtained from the latent class distribution restricted to two values $\rho_{c^\pm}$:

$$\rho_{c^\pm}(c^-) = \frac{\rho(c^-)}{\rho(c^-)+\rho(c^+)}, \; \rho_{c^\pm}(c^+) = \frac{\rho(c^+)}{\rho(c^-)+\rho(c^+)} .$$

Definition 2. We refer to the *supervised data distribution* $\mathcal{S}$ as the process that generates a labelled sample $(\mathbf{x}, y)$ according to the following scheme:
1. Draw a class $c \sim \rho_{c^\pm}$ and set label $y = y_c$;
2. Draw a sample $\mathbf{x} \sim \mathcal{D}_c$.

Loss function. The learning process is divided in two sequential steps, the unsupervised and supervised steps. In order to relate these two steps, the key is to express them in terms of a common convex loss function $\ell : \mathbb{R} \to \mathbb{R}^+$. Typical choices are

$$\ell_{\text{log}}(v) := \log_2(1 + e^{-v}) , \qquad \textit{(logistic loss)} \qquad (1)$$
$$\ell_{\text{hinge}}(v) := \max\{0, 1 - v\} , \qquad \textit{(hinge loss)} \qquad (2)$$

where the loss argument v expresses a notion of *margin*.

In the first step, an unsupervised representation learning algorithm produces a feature map $\mathbf{f} : X \to \mathbb{R}^d$. The *contrastive loss* associated with $\mathbf{f}$ is defined as

$$L_{\text{un}}(\mathbf{f}) := \mathop{\mathbb{E}}_{(c^+,c^-)\sim\rho^2} \; \mathop{\mathbb{E}}_{\substack{(\mathbf{x},\mathbf{x}^+)\sim\mathcal{D}_{c^+}^2 \\ \mathbf{x}^-\sim\mathcal{D}_{c^-}}} \ell\Big(\mathbf{f}(\mathbf{x})\cdot\big[\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-)\big]\Big)$$
$$= \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{x}^+,\mathbf{x}^-)\sim\mathcal{U}} \ell\Big(\mathbf{f}(\mathbf{x})\cdot\big[\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-)\big]\Big) .$$

More precisely, from the unsupervised training dataset

$$U = \{(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)\}_{i=1}^m \sim \mathcal{U}^m , \qquad (3)$$

we are interested in learning the feature map $\mathbf{f}$ that minimises the following empirical contrastive loss:

$$\widehat{L}_{\text{un}}(\mathbf{f}) := \frac{1}{m} \sum_{i=1}^m \ell\Big(\mathbf{f}(\mathbf{x}_i) \cdot \big[\mathbf{f}(\mathbf{x}_i^+) - \mathbf{f}(\mathbf{x}_i^-)\big]\Big) . \qquad (4)$$

In the second step, a supervised learning algorithm is given the *mapped* dataset $\widehat{S} := \{(\hat{\mathbf{x}}_i, y_i)\}_{i=1}^n$, with $\hat{\mathbf{x}}_i := \mathbf{f}(\mathbf{x}_i)$, and returns a predictor $g : \mathbb{R}^d \to \mathbb{R}$. For a fixed pair $c^\pm = \{c^-, c^+\}$, the predicted label on an input $\mathbf{x}$ is then obtained from $\hat{y} = \operatorname{sgn}[g(\hat{\mathbf{x}})]$ (recall that $Y = \{-1, 1\}$), and we aim to minimise the supervised loss

$$L_{\text{sup}}(g \circ \mathbf{f}) := \mathop{\mathbb{E}}_{c\sim\rho_{c^\pm}} \; \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_c} \ell\Big(y_c \, g(\mathbf{f}(\mathbf{x}))\Big)$$
$$= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{S}} \ell\Big(y \, g(\mathbf{f}(\mathbf{x}))\Big) .$$

Given a labelled dataset $S \sim \mathcal{S}^n$, the empirical counterpart of the above supervised loss is

$$\widehat{L}_{\text{sup}}(g \circ \mathbf{f}) := \frac{1}{n} \sum_{i=1}^n \ell\Big(y_i \, g(\mathbf{f}(\mathbf{x}_i))\Big) .$$

Mean classifier. Following Arora et al. (2019), we study the mean classifier defined by the linear function

$$g_{c^\pm}(\hat{\mathbf{x}}) := \mathbf{w}_{c^\pm} \cdot \hat{\mathbf{x}} ,$$

where $\mathbf{w}_{c^\pm} := \boldsymbol{\mu}_{c^+} - \boldsymbol{\mu}_{c^-}$, and $\boldsymbol{\mu}_c := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_c} \mathbf{f}(\mathbf{x})$. Then, the *supervised average loss* of the mean classifier is the expected loss on a dataset whose pair of labels is sampled from the latent class distribution ρ.

$$L^{\mu}_{\text{sup}}(\mathbf{f}) \coloneqq \mathop{\mathbb{E}}_{c^{\pm}\sim\rho^2_{\text{w/o}}} L_{\text{sup}}(g_{c^{\pm}} \circ \mathbf{f}) \,, \tag{5}$$

with $\rho^2_{\text{w/o}}$ being a shorthand notation for the sampling *without replacement* of two classes among $\mathcal{C}$. Indeed, we want positive and negative samples that are generated by distinct latent class distributions, *i.e.*, $c^- \neq c^+$.

2.2 GENERALISATION GUARANTEES

A major contribution of the framework introduced by Arora et al. (2019) is that it rigorously links the unsupervised representation learning task and the subsequent prediction task: it provides generalisation guarantees on the supervised average loss of Eq. (5) in terms of the empirical contrastive loss in Eq. (4). Central to this result is the upcoming Lemma 3, that relates the supervised average loss of the mean classifier to its unsupervised loss.

Lemma 3 (Arora et al., 2019, Lemma 4.3)**.** *Given a latent class distribution ρ on $\mathcal{C}$ and a convex loss $\ell : \mathbb{R}\to\mathbb{R}$, for any feature map $\mathbf{f} : \mathbb{R}^{d_0} \to \mathbb{R}^d$, we have*

$$L^{\mu}_{\text{sup}}(\mathbf{f}) \;\le\; \frac{1}{1-\tau}\left(L_{\text{un}}(\mathbf{f}) - \tau\right),$$

where τ is the probability of sampling twice the same latent class ($\mathbf{1}[\cdot]$ is the indicator function):

$$\tau \;\coloneqq\; \mathop{\mathbb{E}}_{c^{\pm}\sim\rho^2} \mathbf{1}[c^+ = c^-] \;=\; \sum_{c\in\mathcal{C}} [\rho(c)]^2 \,. \tag{6}$$

Arora et al. (2019) upper bound the unsupervised contrastive loss in Lemma 3 by its empirical estimates. The obtained generalisation guarantee is presented by the following Theorem 4. The bound focuses on a class of feature map functions $\mathcal{F}$ through its empirical Rademacher complexity on a training dataset U, defined by

$$\mathcal{R}_U(\mathcal{F}) \coloneqq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}\sim\{\pm1\}^{3dm}} \left(\sup_{f\in\mathcal{F}} \left[\boldsymbol{\sigma} \cdot \mathbf{f}_{|U} \right] \right),$$

where $\mathbf{f}_{|U} \;\coloneqq\; \text{vec}(\{\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_i^+), \mathbf{f}(\mathbf{x}_i^-)\}_{i=1}^m) \;\in\; \mathbb{R}^{3dm}$ is the concatenation of all feature mapping given by $\mathbf{f}$ on U, and $\boldsymbol{\sigma}\sim\{\pm1\}^{3dm}$ denotes the uniformly sampled Rademacher variables over that "representation" space.

Theorem 4 (Arora et al., 2019, Theorem 4.1)**.** *Let $B \in \mathbb{R}_+$ be such that $\|\mathbf{f}(\cdot)\| \le B$, with probability $1-\delta$ over training samples $U \sim \mathcal{U}^m$, $\forall \mathbf{f}\in\mathcal{F}$*

$$L^{\mu}_{\text{sup}}(\widehat{\mathbf{f}}) \le$$
$$\frac{1}{1-\tau}\left(L_{\text{un}}(\mathbf{f}) - \tau\right) + \frac{1}{1-\tau}\,\mathcal{O}\!\left(B\frac{\mathcal{R}_U(\mathcal{F})}{m} + B^2\sqrt{\frac{\ln\frac{1}{\delta}}{m}} \right),$$

where $\hat{\mathbf{f}} \coloneqq \mathop{\text{argmin}}_{\mathbf{f}\in\mathcal{F}} \widehat{L}_{\text{un}}(\mathbf{f})\,.$

3 PAC-BAYES ANALYSIS

Among the different techniques to analyse generalisation in statistical learning theory, PAC-Bayes has emerged in the late 90s as a promising alternative to the Rademacher complexity. PAC-Bayes (pioneered by Shawe-Taylor and Williamson, 1997; McAllester, 1998; Catoni, 2003, 2004, 2007 – see Guedj, 2019 for a recent survey) consists in obtaining PAC (probably approximately correct, Valiant, 1984) generalisation bounds for Bayesian-flavoured predictors. PAC-Bayes bounds typically hold with arbitrarily high probability and express a trade-off between the empirical risk on the training set and a measure of complexity of the predictors class. A particularity of PAC-Bayes is that the complexity term relies on a divergence measure between a prior belief and a data-dependent posterior distribution (typically the Kullback-Leibler divergence).

3.1 SUPERVISED LEARNING FRAMEWORK

Let $\mathcal{P}$ be a prior over a predictor class $\mathcal{H}$, which cannot depend on training data, and let $\mathcal{Q}$ be a posterior over the predictor class $\mathcal{H}$, which can depend on the training data. Any predictor $h \in \mathcal{H}$ is a classification function $h : X \to Y$. Most PAC-Bayes results measure the *discrepancy* between the prior and the posterior distributions through the Kullback-Leibler divergence,

$$\text{KL}(\mathcal{P}\|\mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{h\sim\mathcal{P}} \ln\frac{\mathcal{P}(h)}{\mathcal{Q}(h)} \,. \tag{7}$$

Moreover, PAC-Bayes provides bounds on the expected loss of the predictors under the distribution $\mathcal{Q}$. Let us present the classical supervised setup, where the zero-one loss is used.[1] We refer to this loss as the classification risk, denoted by $r(y,\hat{y}) \coloneqq \mathbf{1}[y\,\hat{y} < 0]$.[2] Given a data-generating distribution $\mathcal{S}$ on $X \times Y$, the expected $\mathcal{Q}$-risk is

$$R(\mathcal{Q}) \coloneqq \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{S}}\;\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} r(y, h(\mathbf{x}))\,,$$

and the empirical counterpart, *i.e.*, the $\mathcal{Q}$-weighted empirical risk on a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim \mathcal{S}^n$, is given by

$$\widehat{R}(\mathcal{Q}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \mathop{\mathbb{E}}_{h\sim\mathcal{Q}} r(y_i, h(\mathbf{x}_i))\,.$$

[1] Classical PAC-Bayes analyses consider the supervised learning setting, but non-supervised learning approaches exist (*e.g.*, Seldin and Tishby, 2010; Higgs and Shawe-Taylor, 2010; Germain et al., 2013).

[2] See Appendix B for a contrastive risk with k negative samples.

The following Theorem 5 expresses an upper bound on the risk $R(\mathcal{Q})$, from the empirical risk $\widehat{R}(\mathcal{Q})$ and the posterior-prior divergence $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$.

Theorem 5 (Catoni, 2007, Theorem 1.2.6)**.** *Given $\lambda > 0$ and a prior $\mathcal{P}$ over $\mathcal{H}$, with probability at least $1-\delta$ over training samples $S \sim \mathcal{S}^n$, $\forall \mathcal{Q}$ over $\mathcal{H}$,*

$$R(\mathcal{Q}) \le \frac{1 - \exp\left(-\lambda \widehat{R}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{n}\right)}{1 - \exp\left(-\lambda\right)}. \tag{8}$$

3.2 PAC-BAYES REPRESENTATION LEARNING

We now proceed to the first of our contributions. We prove a PAC-Bayesian bound on the contrastive unsupervised representation loss, by replacing the Rademacher complexity in Theorem 4 with a Kullback-Leibler divergence. To do so, we consider a prior $\mathcal{P}$ and posterior $\mathcal{Q}$ distributions over a class of feature mapping functions $\mathcal{F} := \{\mathbf{f} \in X \to \mathbb{R}^d\}$. Note that our PAC-Bayesian analysis for a multi-class extension is found at Appendix A.2.

First, let us remark that we can adapt Theorem 5 to a bound on the unsupervised expected *contrastive risk* defined as

$$R_{\mathrm{un}}(\mathcal{Q}) := \underset{(\mathbf{x},\mathbf{x}^+,\mathbf{x}^-)\sim\mathcal{U}}{\mathbb{E}} \; \underset{\mathbf{f}\sim\mathcal{Q}}{\mathbb{E}} r\Big(\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-), \mathbf{f}(\mathbf{x})\Big),$$

where $r(\mathbf{y}, \hat{\mathbf{y}}) := \mathbf{1}[\mathbf{y} \cdot \hat{\mathbf{y}} < 0]$ is the zero-one loss extended to vector arguments. We denote $\widehat{R}_{\mathrm{un}}(\mathcal{Q})$ the empirical counterpart of $R_{\mathrm{un}}(\mathcal{Q})$ computed on the unsupervised training set $U \sim \mathcal{U}^m$. Once expressed this way, Theorem 5—devoted to classical supervised learning—can be straightforwardly adapted for the expected contrastive risk. Thus, we obtain the following Corollary 6.

Corollary 6. *Given $\lambda > 0$ and a prior $\mathcal{P}$ over $\mathcal{F}$, with probability at least $1-\delta$ over training samples $U \sim \mathcal{U}^m$, $\forall \mathcal{Q}$ over $\mathcal{F}$,*

$$R_{\mathrm{un}}(\mathcal{Q}) \le \frac{1 - \exp\left(-\lambda \widehat{R}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{m}\right)}{1 - \exp\left(-\lambda\right)}.$$

Unfortunately, the bound on the contrastive risk $R_{\mathrm{un}}(\cdot)$ does not translate directly to a bound on the supervised average risk

$$R^{\mu}_{\mathrm{sup}}(\mathbf{f}) := \underset{c^{\pm}\sim\rho^2_{\mathrm{w/o}}}{\mathbb{E}} R_{\mathrm{sup}}(g_{c^{\pm}} \circ \mathbf{f}). \tag{9}$$

This is because the zero-one loss is not convex, preventing us from applying Lemma 3 to obtain a result analogous to Theorem 4. However, note that both loss functions defined by Equations (1-2) are upper bound on the zero-one loss:

$$\forall \mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^d : r(\mathbf{y}, \hat{\mathbf{y}}) \le \ell(\mathbf{y} \cdot \hat{\mathbf{y}}), \text{ with } \ell \in \{\ell_{\mathrm{log}}, \ell_{\mathrm{hinge}}\}.$$

Henceforth, we study the $\mathcal{Q}$ expected loss

$$L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) = \underset{\mathbf{f}\sim\mathcal{Q}}{\mathbb{E}} L^{\mu}_{\mathrm{sup}}(\mathbf{f})$$

in regards to

$$L_{\mathrm{un}}(\mathcal{Q}) = \underset{\mathbf{f}\sim\mathcal{Q}}{\mathbb{E}} L_{\mathrm{un}}(\mathbf{f}).$$

By assuming that the representation vectors are bounded, *i.e.*, $\|\mathbf{f}(\cdot)\| \le B$ for some $B \in \mathbb{R}^+$ as in Theorem 4, we also have that the loss function is bounded. Thus, by rescaling in $[0,1]$ the loss function, Theorem 5 can be used to derive the following Theorem 7, which is the PAC-Bayesian doppelgänger of Theorem 4.

Theorem 7. *Let $B \in \mathbb{R}_+$ such that $\|\mathbf{f}(\cdot)\| \le B$ for all $\mathbf{f} \in \mathcal{F}$. Given $\lambda > 0$ and a prior $\mathcal{P}$ over $\mathcal{F}$, with probability at least $1-\delta$ over training samples $U \sim \mathcal{U}^m$, $\forall \mathcal{Q}$ over $\mathcal{F}$,*

$$L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) \le \frac{1}{1-\tau}\left(B_\ell \frac{1 - \exp\left(-\frac{\lambda}{B_\ell}\widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{m}\right)}{1 - \exp(-\lambda)} - \tau\right), \tag{10}$$

with $B_\ell := \max\{\ell(-2B^2), \ell(2B^2)\}$ and τ given by Eq. (6).

Proof. Since $\|\mathbf{f}(\cdot)\| \le B$, we have $\forall \mathbf{x}, \mathbf{x}^+, \mathbf{x}^- \in X^3$:

$$-2B^2 \le \mathbf{f}(\mathbf{x}) \cdot [\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-)] \le 2B^2.$$

Thus, $\ell(\mathbf{f}(\mathbf{x}) \cdot [\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^-)]) \le B_\ell$, as ℓ is both convex and positive. Therefore, the output of the rescaled loss function $\ell'(\cdot) := \frac{1}{B_\ell}\ell(\cdot)$ belongs to $[0,1]$. From that point, we apply Theorem 5 to obtain[3], with probability at least $1-\delta$,

$$\frac{1}{B_\ell} L_{\mathrm{un}}(\mathcal{Q}) \le \frac{1 - \exp\left(-\frac{\lambda}{B_\ell}\widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}}{m}\right)}{1 - \exp\left(-\lambda\right)}.$$

Also, since the inequality stated in Lemma 3 holds true for all $\mathbf{f} \in \mathcal{F}$, taking the expected value according to $\mathcal{Q}$ gives

$$L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) \le \frac{1}{1-\tau}\left(L_{\mathrm{un}}(\mathcal{Q}) - \tau\right).$$

The desired result is obtained by replacing $L_{\mathrm{un}}(\mathcal{Q})$ in the equation above by its bound in terms of $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$. □

[3] Theorem 5 is given for the zero-one loss, but many works show that the same argument holds for any $[0,1]$-bounded loss (*e.g.*, Higgs and Shawe-Taylor, 2010).

The Rademacher bound of Theorem 4 and the PAC-Bayes bound of Theorem 7 convey a similar message: finding a good representation mapping (in terms of the empirical contrastive loss) guarantee to generalise well, on average, on the supervised tasks.

An asset of the PAC-Bayesian bound lies in the fact that its exact value is easier to compute than the Rademacher one. Indeed, for a well-chosen prior-posterior family, the complexity term $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ has a closed-form solution, while computing $\mathcal{R}_U(\mathcal{F})$ involves a combinatorial complexity. From an algorithm design perspective, the fact that $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ varies with $\mathcal{Q}$ suggests a trade-off between accuracy and complexity to drive the learning process, while $\mathcal{R}_U(\mathcal{F})$ is constant for a given choice of class $\mathcal{F}$. We leverage these assets to propose a bound-driven optimisation procedure for neural networks in Section 4.

Note that one could be interested to study the risk of a predictor learned on the representation of the supervised data instead of the mean classifier's risk. As discussed in Appendix A.3, the loss of the best supervised predictor is at least as good as the mean classifier's one.

3.3 RELAXING THE IID ASSUMPTION

An interesting byproduct of Arora et al. (2019)'s approach is that the proof of the main bound (Theorem 4) is modular: we mean that in the proof of Theorem 7, instead of plugging in Catoni's bound (Theorem 5), we can use any relevant bound. We therefore leverage the recent work of Alquier and Guedj (2018) who proved a PAC-Bayes generalisation bound which no longer needs to assume that data are iid, and even holds when the data-generating distribution is heavy-tailed. We can therefore cast our results onto the non-iid setting.

We believe removing the iid assumption is especially relevant for contrastive unsupervised learning, as we deal with triplets of data points governed by a relational causal link (similar and dissimilar examples). In fact, several contrastive representation learning algorithms violate the iid assumption (Goroshin et al., 2015; Logeswaran and Lee, 2018).

Alquier and Guedj (2018)'s framework generalises the Kullback-Leibler divergence in the PAC-Bayes bound with the class of f-divergences (see Csiszár and Shields, 2004, for an introduction). Given a convex function f such that $f(1) = 0$, the f-divergence between two probability distributions is given by

$$D_f(\mathcal{P}\|\mathcal{Q}) = \mathop{\mathbb{E}}_{h\sim\mathcal{Q}} f\left(\frac{\mathcal{P}(h)}{\mathcal{Q}(h)}\right). \qquad (11)$$

Theorem 8. *Given $p > 1, q = \frac{p}{p-1}$ and a prior $\mathcal{P}$ over $\mathcal{F}$, with probability at least $1-\delta, \forall \mathcal{Q}$ over $\mathcal{F}$,*

$$L^{\mu}_{\mathrm{sup}}(\mathcal{Q}) \leq \frac{1}{1-\tau}\left(\widehat{L}_{\mathrm{un}}(\mathcal{Q}) - \tau\right) + \frac{1}{1-\tau}\left(\frac{\mathcal{M}_q}{\delta}\right)^{\frac{1}{q}} \left(D_{\phi_p-1}(\mathcal{Q}\|\mathcal{P}) + 1\right)^{\frac{1}{p}}, \qquad (12)$$

where $\mathcal{M}_q = \mathbb{E}_{\mathbf{f}\sim\mathcal{P}}\,\mathbb{E}_{U\sim\mathcal{U}^m}(|L_{\mathrm{un}}(\mathbf{f}) - \widehat{L}_{\mathrm{un}}(\mathbf{f})|^q)$ (recall that $\widehat{L}_{\mathrm{un}}$ depends on U, see Eqs. 3 and 4) and $\phi_p(x)=x^p$.

The proof is a straightforward combination of aforementioned results, substituting Theorem 1 in Alquier and Guedj (2018) to Catoni's bound (Theorem 5) in the proof of Theorem 7. Up to our knowledge, Theorem 8 is the first generalisation bound for contrastive unsupervised representation learning that holds without the iid assumption, therefore extending the framework introduced by Arora et al. (2019) in a non-trivial and promising direction. Note that Theorem 8 does not require iid assumption for both unsupervised and supervised steps.

4 FROM BOUNDS TO ALGORITHMS

In this section, we propose contrastive unsupervised representation learning algorithms derived from the PAC-Bayes bounds stated in Theorems 7 and 8. The algorithms are obtained by optimising the weights of a neural network by minimising the right-hand side of (10) and (12), respectively. Our training method is inspired by the work of Dziugaite and Roy (2017), who optimise a PAC-Bayesian bound in a supervised classification framework, and show that it leads to non-vacuous bounds values and accurately detects overfitting.

4.1 NEURAL NETWORK OPTIMISATIONS

4.1.1 Algorithm based on Theorem 7

We consider a neural network architecture with N real-valued learning parameters. Let us denote $\mathbf{w} \in \mathbb{R}^N$ the concatenation into a single vector of all the weights, and $\mathbf{f}_{\mathbf{w}} : X \to \mathbb{R}^d$ the output of the neural network whose output is a d-dimensional *representation* vector of its input. From now on, $\mathcal{F}_N = \{\mathbf{f}_{\mathbf{w}} | \mathbf{w} \in \mathbb{R}^N\}$ is the set of all possible neural networks for the chosen architectures. We restrict the posterior and prior over $\mathcal{F}_N$ to be Gaussian distributions, that is

$$\mathcal{Q} := \mathcal{N}(\boldsymbol{\mu}_{\mathcal{Q}}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\mathcal{Q}}))\,, \quad \mathcal{P} := \mathcal{N}(\boldsymbol{\mu}_{\mathcal{P}}, \sigma^2_{\mathcal{P}} I)\,,$$

where $\boldsymbol{\mu}_{\mathcal{Q}}, \boldsymbol{\mu}_{\mathcal{P}} \in \mathbb{R}^N$, $\boldsymbol{\sigma}^2_{\mathcal{Q}} \in \mathbb{R}^N_+$, and $\sigma^2_{\mathcal{P}} \in \mathbb{R}_+$.

Given a fixed λ in Theorem 7, since τ is a constant value, minimising the upper bound is equivalent to minimising the following expression[4]

$$\lambda\, m\, \widehat{L}_{\mathrm{un}}(\mathcal{Q}) + \mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\frac{1}{\delta}\,. \tag{13}$$

Since $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$ is still intractable (as it is expressed as the expectation with respect to the posterior distribution on predictors), we resort to an unbiased estimator; the weight parameters are sampled at each iteration of a gradient descent, according to

$$\mathbf{w} = \boldsymbol{\mu}_{\mathcal{Q}} + \boldsymbol{\sigma}_{\mathcal{Q}} \odot \boldsymbol{\epsilon}\,;\ \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I)\,,$$

the symbol $\odot$ being the element-wise product. Therefore we optimise the posterior's parameters $\boldsymbol{\mu}_{\mathcal{Q}}$ and $\boldsymbol{\sigma}^2_{\mathcal{Q}}$. In addition, we optimise the prior variance $\sigma^2_{\mathcal{P}}$ in the same way as Dziugaite and Roy (2017, Section 3.1). That is, given fixed $b, c \in \mathbb{R}_+$, we consider the bound value for

$$\sigma^2_{\mathcal{P}} \in \left\{c\exp\left(-\tfrac{j}{b}\right) \mid j \in \mathbb{N}\right\}. \tag{14}$$

From the union bound argument, the obtained result is valid with probability $1-\delta$ by computing each bound with a confidence parameter $\delta_j := 1 - \frac{6}{\pi^2 j^2}$, where $j = b\ln\frac{c}{\sigma^2_{\mathcal{P}}}$.

Given δ, b, c, and λ, our final objective based on Theorem 7 is

$$\min_{\boldsymbol{\mu}_{\mathcal{Q}}, \boldsymbol{\sigma}^{\mathbf{2}}_{\boldsymbol{\mathcal{Q}}}, \sigma^2_{\mathcal{P}}} \lambda\, m\, \widehat{L}_{\mathrm{un}}(\mathcal{Q}) + \mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + 2\ln\left(b\ln\tfrac{c}{\sigma^2_{\mathcal{P}}}\right),$$

where

$$\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) = \frac{1}{2}\left(\frac{\|\boldsymbol{\mu}_{\mathcal{Q}}-\boldsymbol{\mu}_{\mathcal{P}}\|_2^2}{\sigma^2_{\mathcal{P}}} - N + \frac{\|\boldsymbol{\sigma}^2_{\mathcal{Q}}\|_1}{\sigma^2_{\mathcal{P}}} + N\ln\sigma^2_{\mathcal{P}} - \sum_{i=1}^{N}\ln\boldsymbol{\sigma}^2_{\mathcal{Q},i}\right).$$

4.1.2 Algorithm based on Theorem 8

We consider the same neural network architecture, prior, and posterior as in Section 4.1.1.

We specify $p = 2$ in Theorem 8 to use a familiar f-divergence: the χ^2-divergence. Then, minimising the upper bound is equivalent to minimising the following expression:

$$\widehat{L}_{\mathrm{un}}(\mathcal{Q}) + \sqrt{\frac{\mathcal{M}_2}{\delta}\left(\chi^2(\mathcal{Q}\|\mathcal{P}) + 1\right)}. \tag{15}$$

Even though we use the unbiased estimator to evaluate the first term like iid algorithm, the objective is still intractable since the moment $\mathcal{M}_2$ requires the test loss $L_{\mathrm{un}}(\mathbf{f})$. Thus we assume the existence of an upper bound on the covariance of the contrastive loss ℓ to bound $\mathcal{M}_2$ as follow[5]:

$$\mathrm{Cov}(\ell(\mathbf{z}_i), \ell(\mathbf{z}_j)) \begin{cases} \leq B_\ell^2 & \text{if } i{-}T \leq j \leq i{+}T \\ = 0 & \text{otherwise} \end{cases}, \tag{16}$$

where T is the length of dependency to generate similarity pairs $(\mathbf{x}, \mathbf{x}^+)$.

This assumption is natural for CURL on sequential data (Mikolov et al., 2013; Goroshin et al., 2015), where a positive sample $\mathbf{x}^+$ appears in sample $\mathbf{x}$'s neighbours in a time series.

Given δ, b, c, and T, our final objective is

$$\min_{\boldsymbol{\mu}_{\mathcal{Q}}, \boldsymbol{\sigma}^2_{\mathcal{Q}}, \sigma^2_{\mathcal{P}}} \widehat{L}_{\mathrm{un}}(\mathcal{Q}) + \pi\left(b\ln\frac{c}{\sigma^2_{\mathcal{P}}}\right)\sqrt{\frac{B_\ell^2}{24m\delta}(1+8T)\left(\chi^2(\mathcal{Q}\|\mathcal{P})+1\right)}, \tag{17}$$

where the full expression of χ^2-divergence is found in Appendix C. The objective is obtained by using the covariance's assumption and the union bound for the prior's variance $\sigma^2_{\mathcal{P}}$.

The objective value is large if T is large, that is when data dependency is long. Therefore collecting independent time-series samples is a more effective way to tighten the bound than increasing T. Interestingly, Eq. (15) with Eq. (16) can be viewed as a generalised bound of Bégin et al. (2016, Corollary 10). In fact, our objective becomes their bound when the data is iid and ℓ is the zero-one loss.

4.2 PARAMETER SELECTION

In the forthcoming experiments (Section 5), we empirically compare the following three criteria for parameter selection: (i) the validation contrastive risk according to the posterior $\mathcal{Q}$, (ii) the validation contrastive risk of the *maximum a posteriori network*, and (iii) the PAC-Bayes bound associated with the learned $\mathcal{Q}$.

For the first validation contrastive risk criterion, we select a model with the best hyper-parameters such that it achieves the lowest contrastive risk $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$ on the validation data. We approximate $\widehat{L}_{\mathrm{un}}(\mathcal{Q})$ in a Monte Carlo fashion by sampling several $\mathbf{f}_{\mathbf{w}}$ from $\mathcal{Q}$.

Empirically, stochastic neural networks learnt by minimising the PAC-Bayes bound perform quite conservatively (Dziugaite and Roy, 2017). Therefore we also use a validation contrastive risk computed with the deterministic neural network being the most likely according to

[4] Note that without loss of generality, the constant B_ℓ is absorbed by λ and plays no role in the optimisation objective.

[5] More generally, we may use α-mixing based upper bound of the moment described by Alquier and Guedj (2018).

the posterior (*i.e.*, the neural network weights are taken as the mean vector of the posterior, rather than sampled from it).

The last criterion, the PAC-Bayes bound, does not use validation data; it only requires training data. For the algorithm described in Section 4.1.1, we select a model with the best hyper-parameters such that it minimises the following PAC-Bayes bound on the contrastive supervised risk $R_{\mathrm{un}}(\mathcal{Q})$:

$$\min_{\lambda>0}\left[\frac{1-\exp\left(-\lambda\widehat{R}_{\mathrm{un}}(\mathcal{Q})-\frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P})+\ln\frac{\pi^2 j^2}{6}+\ln\frac{2\sqrt{m}}{\delta}}{m}\right)}{1-\exp(-\lambda)}\right]. \tag{18}$$

This criterion is given by Corollary 6, where the term $\ln\frac{1}{\delta}$ is replaced by $\ln\frac{\pi^2 j^2}{6}+\ln\frac{2\sqrt{m}}{\delta}$. The first summand comes from the union bound over the prior's variances–see Eq. (14). The second summand replaces $\frac{1}{\delta}$ by $\frac{2\sqrt{m}}{\delta}$, as Letarte et al. (2019, Theorem 3) showed that this suffices to make the bound valid uniformly for all $\lambda > 0$, which allows for minimising the bound over λ. Note that the learning algorithm minimises a bound on the (differentiable) convex loss, but our model selection bound focuses on the zero-one loss as our task is a classification one.

5 NUMERICAL EXPERIMENTS

Our experimental codes are publicly available.[6] We implemented all algorithms with `PyTorch` (Paszke et al., 2019). Herein, we report experiments for the algorithm described in Section 4.1.1. Experiments for the non-iid algorithm are provided in Appendix E.

5.1 PROTOCOL

Datasets. We use `CIFAR-100` (Krizhevsky, 2009) image classification task, containing 60 000 images, equally distributed into 100 labels. We create train/validation/test splits of 47 500/2 500/10 000 images. We preprocess the images by normalising all pixels per channel based on the training data. We build the unsupervised contrastive learning dataset by considering each of the 100 label as a latent class, using a block size of 2 and a number of negative samples of 4 (see Appendix A for the extended theory for block samples and more than one negative samples).

We also use `AUSLAN` (Kadous, 2002) dataset that contains 95 labels, each one being a sign language's motion, and having 22 dimensional features. We split the dataset into 89 775/12 825/12 825 training/validation/test sets. As pre-processing, we normalise feature vectors per dimension based on the training data. The contrastive learning dataset then contains 95 latent classes. The block size and the number of negative samples are the same as `CIFAR-100` setting. More details are provided in Appendix D.1.

Neural networks architectures. For `CIFAR-100` experiments, we use a two hidden convolutional layers neural network (CNN). The two hidden layers are convolutions (kernel size of 5 and 64 channels) with the ReLU activation function, followed by max-pooling (kernel size of 3 and stride of 2). The final layer is a fully connected linear layer (100 neurons) without activation function. For `AUSLAN` experiments, we used a fully connected one hidden layer network with the ReLU activation function. Both hidden and last layers have 50 neurons. More architecture details are given in Appendix D.2.

PAC-Bayes bound optimisation. We learn the network parameters by minimising the bound given by Theorem 7, using the strategy proposed in Section 4.1.1. We rely on the logistic loss given by Eq. (1). We fix the following PAC-Bayes bound's parameters: $b = 100, c = 0.1$, and $\delta = 0.05$. The prior variance is initialised at e^{-8}. The prior mean parameters $\boldsymbol{\mu}_{\mathcal{P}}$ coincide with the random initialisation of the gradient descent.

We repeat the optimisation procedure with different combinations of hyper-parameters. Namely, the PAC-Bayes bound constant λ is chosen in $\{\frac{10^a}{m}|a=1,2,\ldots,9\}$ for `CIFAR-100`, and in $\{\frac{10^a}{m}|a=0,1,\ldots,5\}$ for `AUSLAN`. We also consider as a hyper-parameter the choice of the gradient descent optimiser, here between *RMSProp* (Tieleman and Hinton, 2012) and *Adam* (Kingma and Lei Ba, 2015). The learning rate is in $\{10^{-3}, 10^{-4}\}$. In all cases, 500 epochs are performed and the learning rate is divided by 10 at the 375^{th} epoch. To select the final model among the ones given by all these hyper-parameter combinations, we experiment three parameter selection criteria based on approaches described in Section 4.2, as detailed below.

– *Stochastic validation* (`s-valid`). This metric is obtained by randomly sampling 10 set of network parameters according to the learnt posterior $\mathcal{Q}$, and averaging the corresponding empirical contrastive loss values computed on validation data. The same procedure is used to perform *early-stopping* during optimisation (we stop the learning process when the loss stops decreasing for 20 consecutive epochs).

– *Deterministic validation* (`det-valid`). This metric corresponds to the empirical contrastive loss values

[6] `https://github.com/nzw0301/pb-contrastive`

computed on validation data of the deterministic network $\mathbf{f}^*$, which corresponds to the mean parameters of the posterior (*i.e.*, the *maximum a posteriori* network given by $\mathcal{Q}$). Early stopping is performed in the same way as for `s-valid`.

– *PAC-Bayes bound* (`PB`). The bound values of the learnt posterior $\mathcal{Q}$ are computed by using Eq. (18). Note that since this method does not require validation data, we perform optimisation over the union of the validation data and the training data. We do not perform early stopping since the optimised objective function is directly the parameter selection metric.

Benchmark methods. We compare our results with two benchmarks, described below (more details are provided in Appendix D.3)

– *Prior contrastive unsupervised learning* (Arora et al., 2019). Following the original work, we minimise the empirical contrastive loss $\widehat{L}_{\text{un}}(\mathbf{f})$. Hyper-parameter selection is performed on the validation dataset as for `s-valid` and `det-valid` described above.

– *Supervised learning* (`supervised`). We also train the neural network in a supervised way, using the label information; Following the experiment of Arora et al. (2019), we add a prediction linear layer to our architectures (with 100 output neurons for `CIFAR-100`, and 95 output neurons for `AUSLAN`), and minimise the multi-class logistic loss function

$$\ell_{\log}(\mathbf{v}) := \log_2(1 + \textstyle\sum_{i=1}^{|Y|} e^{-v_i})\,.$$

Once done, we drop the prediction layer. Then, we use the remaining network to extract feature representation.

5.2 EXPERIMENTAL RESULTS

Supervised classification. Table 1 contains supervised accuracies obtained from the representation learnt with the two benchmark methods, as well as with our three parameter selection strategies on the PAC-Bayes learning algorithms. For each method, two types of supervised predictor are used: μ and μ-5 (as in Arora et al., 2019).[7] The μ classifier is obtained $\boldsymbol{\mu}_c$ that was the average vector of feature vectors $\hat{\mathbf{f}}_{\mathbf{w}}$ mapped from training data per supervised label, and μ-5 classifier had $\boldsymbol{\mu}_c$ that was average of 5 random training samples feature vectors. For μ-5, we used averaged evaluation scores over 5 times samplings on each experiment.

For the two datasets, we report three accuracies on the testing set, described below. Values are calculated by averaging over three repetitions of the whole experiments using different random seeds.

– *predictors-2 accuracy* (`AVG-2`). This is the empirical counterpart of Eq. (9), *i.e.*, given a test dataset $T := \{(\mathbf{z}_i, c_i)\}_{i=1}^{|T|}$ where $c_i \in \mathcal{C}$ is a latent class, we define $\texttt{AVG-2}(\mathbf{f}_{\mathbf{w}}) := 1 - \widehat{R}^{\mu}_{\sup}(\mathbf{f}_{\mathbf{w}})$, given

$$\widehat{R}^{\mu}_{\sup}(\mathbf{f}_{\mathbf{w}}) := \frac{C(C-1)}{2} \sum_{1\leq c^+ < c \leq C} \widehat{R}_{T_{c^\pm}}(\hat{g}_{c^\pm} \circ \hat{\mathbf{f}}_{\mathbf{w}})\,,$$

where C is the number of latent classes (*e.g.*, C=100 for `CIFAR-100` dataset), $\hat{\mathbf{f}}_{\mathbf{w}}$ is a feature map learnt from the training data, $\hat{g}_{c^\pm}(\hat{\mathbf{x}}) := (\hat{\mathbf{w}}_{c^+} - \hat{\mathbf{w}}_{c^-}) \cdot \hat{\mathbf{x}}$ is the predictor based on the centre of mass $\hat{\mathbf{w}}_{c^+}, \hat{\mathbf{w}}_{c^-}$ of the training data mapped features of classes c^+, c^-, and $\widehat{R}_{T_{c^\pm}}$ is the supervised risk on the dataset $T_{c^\pm} := \{(\mathbf{x}, 1) | (\mathbf{x}, c^+) \in T\} \cup \{(\mathbf{x}, -1) | (\mathbf{x}, c^-) \in T\}$:

$$\widehat{R}_{T_{c^\pm}}(\hat{g}_{c^\pm} \circ \hat{\mathbf{f}}_{\mathbf{w}}) := \frac{1}{|T_{c^\pm}|} \sum_{(\mathbf{x},y)\in T_{c^\pm}} r\big(\hat{g}_{c^\pm}(\hat{\mathbf{f}}_{\mathbf{w}}(\mathbf{x})), y\big)\,.$$

– *Top-1 accuracy* (`TOP-1`). This is the accuracy on the multi-class labelled test data T. We predicted the label $\hat{y}_i = \text{argmax}_y\, \boldsymbol{\mu}_y \cdot \mathbf{f}_{\mathbf{w}}(\mathbf{x}_i)$ on the test data. Therefore,

$$\texttt{TOP-1}(\mathbf{f}_{\mathbf{w}}) := \frac{1}{|T|} \sum_{i=1}^{|T|} \mathbf{1}[y_i = \hat{y}_i]\,.$$

– *Top-5 accuracy* (`TOP-5`). For each test instance $(\mathbf{x}_i, y_i) \in T$, let $\hat{Y}_i$ be the set of 5 labels having the highest inner products $\boldsymbol{\mu}_y \cdot \mathbf{f}(\mathbf{x}_i)$. Then,

$$\texttt{TOP-5}(\mathbf{f}_{\mathbf{w}}) := \frac{1}{|T|} \sum_{i=1}^{|T|} \mathbf{1}[y_i \in \hat{Y}_i]\,.$$

Note that the `TOP-1` and `TOP-5` metrics are not supported by theoretical results, in the present paper or the work of Arora et al. (2019). Nevertheless, we report those as an empirical hint of how representations are learnt by our contrastive unsupervised representation learning algorithm.

We observe that `det-valid` algorithm achieves competitive results with the ones of the CURL algorithm studied by Arora et al. (2019).

PAC-Bayesian generalisation bounds. Table 2 shows the PAC-Bayes bound values obtained from Eq. (18). The bounds were calculated by using the same models used in Table 1. We also reported a training risk $\widehat{R}_{\text{un}}(\mathbf{f}^*)$ and test risk $R_{\text{un}}(\mathbf{f}^*)$ that we calculated by using only the mean parameter of the posterior as for neural network's weight. The rows of $\widehat{\lambda}$ indicated the optimised λ

[7] Our neural network architecture on `CIFAR-100` differs from the one used in Arora et al. (2019). Their model is based on the deeper network `VGG-16` (Simonyan and Zisserman, 2015), which explains why our accuracies are lower than the one reported in Arora et al. (2019).

Table 1: Supervised tasks results. supervised was trained on the labelled training data, the others were trained on the contrastive training data. For supervised, Arora et al. (2019), s-valid, and det-valid, hyper-parameters were selected by using the validation loss. PB hyper-parameters were selected by the PAC-Bayes bound. The best scores are in bold among contrastive learning algorithms.

					PAC-Bayes based methods					
	supervised		Arora et al. (2019)		s-valid		det-valid		PB	
	μ	μ-5	μ	μ-5	μ	μ-5	μ	μ-5	μ	μ-5
CIFAR-100										
AVG-2	91.4	87.5	89.4	85.6	87.7	83.9	**90.0**	**87.2**	75.4	70.8
TOP-1	25.3	16.8	**22.5**	15.6	17.3	12.7	21.4	**16.0**	6.9	5.4
TOP-5	57.8	46.0	52.9	42.6	46.9	38.3	**54.0**	**45.2**	23.4	19.4
AUSLAN										
AVG-2	80.2	75.1	**85.6**	**83.3**	85.3	82.7	85.3	82.9	82.6	79.1
TOP-1	12.0	7.1	**38.0**	**24.9**	36.1	23.7	37.1	24.7	23.2	14.8
TOP-5	35.7	24.1	**56.7**	48.2	56.2	47.7	56.5	**49.1**	50.6	38.4

Table 2: Contrastive unsupervised PAC-Bayes bounds of the models used in Table 1.

	s-valid	det-valid	PB
CIFAR-100			
$\widehat{R}_{\text{un}}(\mathbf{f}^*)$	0.146	0.131	0.308
$R_{\text{un}}(\mathbf{f}^*)$	0.185	0.167	0.315
$\widehat{R}_{\text{un}}(\mathcal{Q})$	0.172	0.170	0.323
$R_{\text{un}}(\mathcal{Q})$	0.203	0.197	0.327
Bound	0.733	0.718	0.437
KL	32 756	30 894	1 333
$\lambda \times m$	10^5	10^5	10^4
$\widehat{\lambda} \times m$	122 781	119 687	24 295
AUSLAN			
$\widehat{R}_{\text{un}}(\mathbf{f}^*)$	0.193	0.190	0.263
$R_{\text{un}}(\mathbf{f}^*)$	0.182	0.182	0.216
$\widehat{R}_{\text{un}}(\mathcal{Q})$	0.199	0.195	0.267
$R_{\text{un}}(\mathcal{Q})$	0.186	0.185	0.220
Bound	0.419	0.417	0.361
KL	9 769	10 018	2 054
$\lambda \times m$	10^5	10^5	10^4
$\widehat{\lambda} \times m$	95 683	97 379	45 198

values that minimised Eq. (18), and thus that correspond to the reported PAC-Bayes bounds. Let us stress that all reported bounds values are non-vacuous.

The generalisation bounds obtained with the PB parameter selection criterion are naturally the tightest. For this method, the gap between the empirical risk $\widehat{R}_{\text{un}}(\mathcal{Q})$ and the test risk $\widehat{R}_{\text{un}}(\mathcal{Q})$ is remarkably consistently small. This highlights that the PAC-Bayesian bound minimisation is not prone to overfitting. On the downside, this behaviour seems to promote "conservative" solutions, which in turns gives lower supervised accuracy compared to methods relying on a validation set (see Table 1).

6 CONCLUSION

We extended the framework introduced by Arora et al. (2019), by adopting a PAC-Bayes approach to contrastive unsupervised representation learning. This allows in particular to (i) derive new algorithms, by minimising the bounds (ii) remove the iid assumption. While supported by novel generalisation bounds, our approach is also validated on numerical experiments are the bound yields non-trivial (non-vacuous) values.

Acknowledgements

We thank Mikhail Khodak and Nikunj Saunshi for sharing their experimental setting and the reviewers for their fruitful comments, and Louis Pujol, Ikko Yamane, and Han Bao for helpful discussions. This work was supported by the French Project APRIORI ANR-18-CE23-0015. KN is supported by JSPS KAKENHI Grant Number 18J20470. PG is supported by the Canada CIFAR AI Chair Program.

References

Pierre Alquier and Benjamin Guedj. Simpler PAC-Bayesian Bounds for Hostile Data. *Machine Learning*, 107(5):887–902, 2018.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In *ICML*, pages 5628–5637, 2019.

Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. PAC-Bayesian Bounds based on the Rényi Divergence. In *AISTATS*, pages 435–444, 2016.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Hans-Hermann Bock. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*, chapter 8.2.2.3 The χ^2-divergence. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, 2000.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018.

Olivier Catoni. A PAC-Bayesian Approach to Adaptive Classification. *preprint*, 2003.

Olivier Catoni. *Statistical Learning Theory and Stochastic Optimization: Ecole d'Eté de Probabilités de Saint-Flour XXXI-2001*. Springer, 2004.

Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *IMS Lecture Notes - Monograph Series*. Institute of Mathematical Statistics, 2007.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv:2002.05709v1*, 2020.

Imre Csiszár and Paul C. Shields. Information Theory and Statistics: A Tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4): 417–528, 2004.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data. In *UAI*, 2017.

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers. In *ICML*, pages 738–746, 2013.

Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised Learning of Spatiotemporally Coherent Metrics. In *ICCV*, pages 4086–4093, 2015.

Benjamin Guedj. A Primer on PAC-Bayesian Learning. *arXiv preprint arXiv:1901.05353v3*, 2019.

Matthew Higgs and John Shawe-Taylor. A PAC-Bayes Bound for Tailored Density Estimation. In *ALT*, pages 148–162, 2010.

Mohammed Waleed Kadous. *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series*. PhD thesis, University of New South Wales, 2002.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Gaël Letarte, Pascal Germain, Benjamin Guedj, and François Laviolette. Dichotomize and Generalize: PAC-Bayesian Binary Activated Deep Neural Networks. In *NeurIPS*, 2019.

Lajanugen Logeswaran and Honglak Lee. An Efficient Framework for Learning Sentence Representations. In *ICLR*, 2018.

David A. McAllester. Some PAC-Bayesian Theorems. In *COLT*, pages 230–234, 1998.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS*, 2013.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. In *ICLR*, 2018.

Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *ECCV*, 2016.

Adam Paszke et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019.

Yevgeny Seldin and Naftali Tishby. PAC-Bayesian Analysis of Co-clustering and Beyond. *Journal of Machine Learning Research*, 11:3595–3646, 2010.

John Shawe-Taylor and Robert C. Williamson. A PAC Analysis of a Bayesian Estimator. In *COLT*, pages 2–9, 1997.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5—rmsprop: Divide the Gradient by a Running Average of its Recent Magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012.

Leslie G. Valiant. A Theory of the Learnable. In *STOC*, pages 436–445, 1984.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization. In *ECCV*, pages 649–666, 2016.