PACTran: PAC-Bayesian Metrics for Estimating the Transferability of Pretrained Models to Classification Tasks

Nan Ding, Xi Chen, Tomer Levinboim, Soravit Changpinyo, Radu Soricut

Google Research
{dingnan,chillxichen,tomerl,schangpi,rsoricut}@google.com

Abstract. With the increasing abundance of pretrained models in recent years, the problem of selecting the best pretrained checkpoint for a particular downstream classification task has been gaining increased attention. Although several methods have recently been proposed to tackle the selection problem (e.g. LEEP, H-score), these methods resort to applying heuristics that are not well motivated by learning theory. In this paper we present PACTran, a theoretically grounded family of metrics for pretrained model selection and transferability measurement. We first show how to derive PACTran metrics from the optimal PAC-Bayesian bound under the transfer learning setting. We then empirically evaluate three metric instantiations of PACTran on a number of vision tasks (VTAB) as well as a language-and-vision (OKVQA) task. An analysis of the results shows PACTran is a more consistent and effective transferability measure compared to existing selection methods.

1 Introduction

Recent advances in machine learning and neural networks have resulted in effective but extremely over-parameterized models [37, 9], sometimes referred to as foundation models [6]. Despite the fact that their training recipe and data are often available, training such models requires access to computational resources that are well beyond the reach of an average machine learning user or group. At the same time, many such model checkpoints (parameter snapshots at a particular training step) have been made publicly available in platforms such as Tensorflow-Hubs[⋆] [1] and Huggingface[⋆⋆] [47], so that ML users who are interested in a certain model configuration need only write a few lines of code to initialize their own model from a public checkpoint, and continue fine-tuning on their downstream task of interest without incurring the cost of pretraining a model themselves. However, as the number of such models and checkpoints increases, a natural question of selection arises – is it possible to tell which initialization checkpoint is most suitable for a given downstream task without brute-force fine-tuning from all the available checkpoints?

[⋆] https://www.tensorflow.org/hub

[⋆⋆] https://huggingface.co/

To answer this question in the context of classification tasks, a number of existing approaches have been recently proposed. For example, the LEEP metric [33] assumes the pretrained model was trained on a source classification task, and then estimates the likelihood of an empirical predictor which maps source labels to target (downstream) labels for classifying the target data. On the other hand, the H-score metric [3] casts the classification problem as a linear regression task involving the representations of the penultimate layer and the target labels.

Unfortunately, both these methods resort to heuristics or approximations to arrive at their estimate. Specifically, the empirical predictor used by LEEP is not the optimal solution to its associated objective function. Furthermore, the predictor and the metric are estimated on the same dataset, which is prone to overfitting. On the other hand, the least squares solution of H-score is generally not a valid approximation to the commonly-used cross-entropy loss for classification, unless the dependence between the label and the input feature is weak [23], which rarely holds in practice.

In this paper we present PACTran, a theoretically grounded framework for deriving metrics that measure the transferability of pretrained models to downstream classification tasks. Our framework seeks an optimal yet efficient PAC-Bayesian bound [31, 16] to the generalization error in a transfer learning setting, and the error is based on the cross-entropy loss between the prediction and the labels, as is commonly used in classification. That is, the PACTran framework enjoys at least one of two advantages compared to previous methods: (1) It is based on learning theory (as opposed to LEEP) through PAC-Bayesian bounds that measure the generalization gap, and (2) it is compatible with classification, since it relies on the cross-entropy loss (as opposed to H-score).

We instantiate the PACTran framework with three different priors, yielding three new transferability metrics: PACTran-Dirichlet, PACTran-Gamma and PACTran-Gaussian. Our experiments empirically evaluate and compare these new metrics against a number of baseline metrics over various image classification tasks in the Visual Task Adaptation Benchmark (VTAB) [51]. Furthermore, we also evaluate our metrics over the multimodal Open-Knowledge VQA [19, 30] task, which contains both image and text.

2 Transferability Metrics: A Quick Review

In this section, we describe the transferability problem and review several transferability metrics which will be used as baselines in our experiments. We begin by describing the setup of Transfer Learning, where the goal is to leverage knowledge acquired on a source task in order to solve a new target task. Specifically, let M denote a model checkpoint already pretrained to solve a source task, and let S denote a dataset for the target (downstream) task, such that $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_N)\}$ with inputs $\mathbf{x} \in \mathcal{X}$ and target labels $y \in \mathcal{Y}$. Transfer learning seeks to transfer the knowledge already encoded in M by finetuning from M using S.

At the same time, given multiple pretrained checkpoints M, another problem arises – is it possible to know which of the pretrained checkpoints is most suitable for the downstream task S, without incurring the cost of fine-tuning from each of them? To this end, several effective and computationally efficient transferability metrics have been proposed and are summarized below.

2.1 LEEP

Given a checkpoint M of a model pretrained on a source classification task, LEEP [33] estimates the transferability of M to the target dataset S by first computing two types of probabilities: (1) the predicted distribution $M(\mathbf{x}_i)$ over the source label set $\mathcal{Z}$ of the pretraining task, where we let $M(\mathbf{x}_i)_z$ denote the output probability of the z-th source label, and (2) the empirical conditional distribution $\hat{p}(y|z)$ of the target label y given the source label z,

$$\hat{p}(y|z) = \frac{\hat{p}(y,z)}{\sum_y \hat{p}(y,z)}, \text{where } \hat{p}(y,z) = \frac{1}{N} \sum_{(x_i,y_i)\in S} M(\mathbf{x}_i)_z \cdot \delta(y_i = y). \tag{1}$$

The LEEP measure is then defined as the logarithm of the marginal likelihood $\hat{p}(\mathbf{y}\,|\,\mathbf{x})$ (called EEP) given the empirical predictor $\hat{p}(y|z)$ and $M(\mathbf{x})$,

$$R_{LEEP} = \frac{1}{N} \sum_{(x_i,y_i)\in S} \log \hat{p}(y_i|\,\mathbf{x}_i) = \frac{1}{N} \sum_{(x_i,y_i)\in S} \log \left(\sum_{z\in\mathcal{Z}} \hat{p}(y_i|z) M(\mathbf{x}_i)_z \right). \tag{2}$$

LEEP was proposed as an improvement over the Conditional Entropy (CE) measure [42] which itself is an information theoretic approach that measures the transferability between two classification tasks by analyzing the correlation between their label sequences $Y = \{y_1, \ldots, y_N\}$ and $Z = \{z_1, \ldots, z_N\}$. In [33] the authors show that LEEP is an upper bound of negative CE and outperforms it empirically as a transferability metric. However, from a theoretic stand point the LEEP formulation suffers from a few deficiencies. For example, plugging in the empirical conditional distribution $\hat{p}(y|z)$ into Eq.(2) is not guaranteed to maximize the target log-likelihood $\log \hat{p}(\mathbf{y}\,|\,\mathbf{x})$. Furthermore, both $\hat{p}(y|z)$ and $\log \hat{p}(\mathbf{y}\,|\,\mathbf{x})$ are computed over S, which make the latter prone to overfitting and behave more similarly to training error as opposed to generalization error.

2.2 $\mathcal{N}$-LEEP

Another limitation of the LEEP measure (as well as CE) is that it can only be applied to measure the transferability of pretrained classification models. In addition, LEEP's performance degrades when the number of source classes is considerably smaller than the number of target classes. To overcome these issues, several methods that propose using the outputs $f(\mathbf{x})$ of the penultimate layer.

In $\mathcal{N}$-LEEP [29], the authors suggest to first apply Principal Component Analysis (PCA) on the penultimate layer outputs $f(\mathbf{x})$ to reduce their dimension and then fit a Gaussian Mixture Model (GMM) to the PCA-reduced representation $\mathbf{s}(\mathbf{x})$, so that $p(\mathbf{s}) = \sum_{v\in\mathcal{V}} \alpha_v \mathcal{N}(\mathbf{s}\,|\,\mathbf{u}_v, \mathbf{\Sigma}_v)$ and the posterior of the cluster assignment

$$p(v|\,\mathbf{x}) = p(v|\,\mathbf{s}) \propto \alpha_v \mathcal{N}(\mathbf{s}\,|\,\mathbf{u}_v, \mathbf{\Sigma}_v) \tag{3}$$

are used to replace $M(\mathbf{x})_z$ in Eq.(1). The rest of the procedure follows the same as in Eq. (2).

The $\mathcal{N}$-LEEP method [29] conjectures that the cluster assignment $p(v|\,\mathbf{s})$ is more reliable than the class assignment $M(\mathbf{x})_z$, because the GMM fitting is learned from the downstream target data, while the softmax classifier of LEEP is learned over the pretrained source data. Since $\mathcal{N}$-LEEP is a extension of LEEP, it also inherits its aforementioned problems such as the non-optimality of the log-likelihood, as well as the lack of generalization consideration.

2.3 H-Score

The H-score [3] transferability metric is also not restricted to pretrained classifiers. The idea for H-score comes from the matrix factorization of the divergence transition matrix (DTM) $\tilde{B} = \frac{p(x,y)}{\sqrt{p(x)p(y)}} - \sqrt{p(x)p(y)}$, for discrete random variables $\mathbf{x}$ and $\mathbf{y}$. It is shown in [23] that, under the assumption of sufficiently small $\tilde{B}$, the solution of the cross-entropy loss coincides with the following solution of the matrix decomposition:

$$\Psi^* = \underset{\Psi}{\operatorname{argmin}} \, \|\tilde{B} - \Phi(\mathbf{x})^\top \Psi\|_F^2, \text{ where } \Phi(\mathbf{x}) = \sqrt{p(\mathbf{x})} f(\mathbf{x}). \tag{4}$$

After plugging in the least squares solution $\Psi^* = \tilde{B}\Phi(\Phi^\top\Phi)^{-1}$, Eq.(4) becomes $\|\tilde{B}\|_F^2 - \|\tilde{B}\Phi(\Phi^\top\Phi)^{-\frac{1}{2}}\|_F^2$, in which the second term is defined as the H-score:

$$H = \|\tilde{B}\Phi(\Phi^\top\Phi)^{-\frac{1}{2}}\|_F^2 = \operatorname{tr}(\operatorname{cov}(f(\mathbf{x})))^{-1}\operatorname{cov}(\mathbb{E}_{p(x|y)}[f(\mathbf{x})|y]). \tag{5}$$

Compared to LEEP, H-score is more theoretically solid, in that it is optimal with respect to its loss. However, the key drawback of the H-score is that the optimality is based on the least squares objective, which is rarely used for classification. As proven in [23], the least squares solution is a valid approximation to the cross-entropy classification loss only when label $\mathbf{y}$ and input $\mathbf{x}$ are weakly dependent, which is clearly not the case in general.

2.4 LogME

Similarly to H-Score, the Log Marginal Evidence (LogME) [49] transferability metric also uses a least squares objective function. However, to avoid overfitting, instead of directly minimizing the Gaussian based log-likelihood (a.k.a the squared-loss) $\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \|\,\mathbf{y} - \mathbf{f}^\top\,\mathbf{w}\,\|_F^2$, LogME uses Bayesian averaging to improve its generalization ability. That is, the LogME metric uses the marginal evidence of the target task $p(y|\,\mathbf{f}) = \int p(\mathbf{w})p(y|\,\mathbf{f}, \mathbf{w})d\,\mathbf{w}$. When $p(\mathbf{w})$ is defined as a Gaussian prior and $p(y|\,\mathbf{f}, \mathbf{w})$ is a Gaussian likelihood, then $p(y|\,\mathbf{f})$ can be analytically estimated.

LogME shares the same theoretical problems as H-score due to its dependence on the least squares objective, however, by relying on the marginal evidence it is less prone to overfitting [49] which potentially improves its generalization ability.

3 PACTran

In this section, we first briefly review the PAC-Bayesian bound [31, 17] in the supervised learning setting. We then show how to leverage this bound for measuring transferability as the PACTran metric (Section 3.1). Specifically, we derive three instances of the PACTran metric based on the cross entropy loss using three different prior distributions: two based on conjugate priors with the Dirichlet and Gamma distributions (Section 3.2 and 3.3), and a third with a non-conjugate Gaussian prior (Section 3.4).

3.1 PAC-Bayesian Bounds for Supervised and Transfer Learning

Consider a learning task with data distribution D where examples are denoted as $u = (x, y)$. A hypothesis h from the hypothesis space H allows us to make predictions for each input x. The quality of the predictions is measured by a loss function $l(h, u)$, and the goal is to minimize the expected loss $L(h, D) = \mathbb{E}_{u\sim D}\, l(h, u)$. Typically, the data distribution D is unknown, and instead we are given a set of N (training) examples $S \sim D^N = \{u_i \sim D\}_{i=1}^N$, in which case the empirical error on S is simply $\hat{L}(h, S) = \frac{1}{N}\sum_{i=1}^N l(h, u_i)$. The gap between $L(h, D)$ and $\hat{L}(h, S)$ is known as the generalization gap of h. Based on this, various forms of PAC (Probably Approximately Correct) bounds have been studied in the ML community over the last few decades [7, 43].

A key drawback of the PAC bounds is that the worst-case analysis (via the union bound over all $h \in H$) makes the bound vacuous for modern machine learning approaches [25, 52]. To address this drawback, PAC-Bayesian learning [31, 17] goes one step further by bounding the generalization gap of distributions over H, which can be optimized to obtain a non-vacuous bound [14, 25]. In particular, let us assume that the learner has some prior knowledge of the hypothesis space H in the form of a prior distribution $P(h)$. Once the learner observes a training dataset S, it updates its prior P into a posterior distribution Q. The expected error of the posterior Q is called the Gibbs error $L(Q, D) = \mathbb{E}_{h\sim Q}\, L(h, D)$, and its empirical counterpart is $\hat{L}(Q, S) = \mathbb{E}_{h\sim Q}\, \hat{L}(h, S)$. The PAC-Bayesian framework provides the following upper bound [31, 16] over $L(Q, D)$ based on its empirical estimate $\hat{L}(Q, S)$:

Theorem 1. *[16] Given a data distribution D, a hypothesis space H, a prior P, a confidence level $\delta \in (0, 1]$, and $\lambda > 0$, with probability at least $1 - \delta$ over* *samples $S \sim D^N$, for all posterior Q,*

$$L(Q,D) \leq \hat{L}(Q,S) + \frac{1}{\lambda} D_{KL}(Q\|P) + C(\delta,\lambda,N) \tag{6}$$

where $C(\delta,\lambda,N)$ is a constant independent of the posterior Q.

The hyperparameter λ can be adjusted to balance between the divergence and the constant C terms, where a common choice is $\lambda \propto N$ (see [16, 38, 11]).

In the transfer learning setting, starting from a pretrained checkpoint M that is encoded within the prior $P(h)$, $L(Q,D)$ measures the generalization error of a posterior Q after it was finetuned over the downstream data S. Furthermore, by minimizing the RHS of the bound (Eq. (6)) with respect to $Q \in \mathcal{Q}_M$, one can obtain a posterior Q that has low transfer error $L(Q,D)$. Therefore, to measure the transferability of a pretrained checkpoint M, we define a family of metrics *PACTran* by optimizing the PAC-Bayesian bound (ignoring the constant C since it is the same for all checkpoints):

$$\min_{Q \in \mathcal{Q}_M} \hat{L}(Q,S) + \frac{1}{\lambda} D_{KL}(Q\|P). \tag{7}$$

For computational efficiency, we restrict the domain of $\mathcal{Q}_M$ in which the feature network of the pretrained checkpoint M remains fixed. Since all $h \in dom(\mathcal{Q}_M)$ shares the same feature network, we can simplify P as the prior distribution of the top classification layer of the network only; and Q as the posterior distribution of the top layer after finetuning. Despite this restriction, PACTran appears promising in comparing the transferability of pretrained checkpoints even after full-model finetuning.

According to [16], the so-called Gibbs posterior Q^* that minimizes the objective of Eq.(7) takes the form of $Q^*(h) = P(h)\exp(-\lambda\hat{L}(h,S))/Z(S)$, where $Z(S)$ is equal to the marginal evidence $\int P(h)\exp(-\lambda\hat{L}(h,S))dh$. Plugging in $Q^*(h)$ back into Eq.(7), the resulting optimal PAC-Bayesian bound equals to $-\frac{1}{\lambda}\log Z(S)$. Note however, that computing $\log Z(S)$ is only analytically feasible, when the prior $P(h)$ and the likelihood function $\exp(-\lambda\hat{L}(h,S))$ are conjugate, for example, when both are Gaussians as in LogME [49].

In this paper, we focus on metrics for which $\hat{L}(h,S)$ is based on the cross-entropy loss, as it is more compatible with classification tasks (in which case $\exp(-L(Q,D))$ is an estimate of the expected test accuracy). From a theoretical perspective, this makes the PACTran metric preferable to LEEP (which is not optimal over the cross entropy loss) as well as LogME and H-score metrics (whose proposed solution is based on the squared loss instead of the classification loss of the downstream task).

In what follows, we derive three instantiations of the bound using the conjugate Dirichlet and Gamma priors whose solution can be found with a fast variational approach, and the non-conjugate Gaussian prior, which requires gradient optimization.

3.2 PACTran with a Dirichlet Prior

Given the target data $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \in (\mathcal{X}, \mathcal{Y})$, let us assume that the pretrained model M provides a probability vector $M(\mathbf{x})$ where $\sum_z M(\mathbf{x})_z = 1$. Here, $z \in \mathcal{Z}$ can either be defined over the set of source label as in LEEP or over the Gaussian clusters as in $\mathcal{N}$-LEEP. We restrict the top layer to a set of l_1-normalized vectors $\mathbf{W} = \left\{\mathbf{w}_1, \ldots, \mathbf{w}_{|\,\mathcal{Z}\,|}\right\}$ in the probability simplex, where for each vector $\mathbf{w}_z$ we have $\sum_{y \in \mathcal{Y}} w_{yz} = 1, w_{yz} \geq 0$, and then define the marginal likelihood as:

$$p(y_i |\, \mathbf{x}_i, \mathbf{W}) = \sum_z p(y_i, z |\, \mathbf{x}_i, \mathbf{W}) = \sum_z M(\mathbf{x}_i)_z w_{y_i z}. \tag{8}$$

We assign a Dirichlet prior $P(\mathbf{w}_z)$ on these vectors and let $\lambda = N$ for simplicity. Using the above definitions, we can rewrite $\log Z(S)$ as

$$\begin{aligned} &\log \int \prod_z P(\mathbf{w}_z) \prod_i \left(\sum_z p(y_i, z_i = z |\, \mathbf{x}_i, \mathbf{W}) \right) d\,\mathbf{W} \\ =&\log \sum_{z_1} \cdots \sum_{z_N} \int \left(\frac{\prod_z \Gamma(\sum_y \alpha_y)}{\prod_z \prod_y \Gamma(\alpha_y)} \right) \prod_z \prod_y w_{yz}^{n_{yz}+\alpha_y-1} \prod_i M(\mathbf{x}_i)_{z_i} d\,\mathbf{W}, \end{aligned} \tag{9}$$

where $\Gamma(\cdot)$ is the well-known Gamma function. The form of the resulting model is similar to Latent Dirichlet Allocation (LDA) [5]. Evaluating Eq.(9) exactly is considered intractable, as it involves a summation over $\mathbf{z}$ which has $|\,\mathcal{Z}\,|^N$ different configurations. Therefore, we turn to the variational inference approach as in [5, 4] to optimize the evidence lower bound (ELBO). The PACTran-Dirichlet is the negation of the optimal ELBO, and equals to (see details in A.1):

$$\sum_z \left(\log C(\tilde{\boldsymbol{\alpha}}_z) - \log C(\boldsymbol{\alpha}_z) + \sum_i q^*(z_i = z) \left(\log q^*(z_i = z) - \log M(\mathbf{x}_i)_z \right) \right), \tag{10}$$

where,

$$\begin{aligned} &q^*(z_i = z) = \text{softmax} \left(\log M(\mathbf{x}_i)_z + \Psi(\tilde{\alpha}_{y_i z}) - \Psi(\sum_y \tilde{\alpha}_{yz}) \right), \\ &\tilde{\alpha}_{yz} = \alpha_{yz} + \sum_i q^*(z_i = z) \delta(y_i = y), \text{ and } \; C(\boldsymbol{\alpha}_z) = \frac{\Gamma(\sum_y \alpha_{yz})}{\prod_y \Gamma(\alpha_{yz})}, \end{aligned}$$

where $\Psi(\cdot)$ denotes the digamma-function.

It is worth noting that the PACTran-Dirichlet metric Eq.(10) is a valid PAC-Bayesian upper bound to the generalization error (up to a constant). That is because Eq.(10) is the negation of ELBO which upper bounds the negative log evidence $-\log Z(S)$ which itself is an upper bound of $L(Q^*, D)$. Furthermore, both upper bounds are optimally tight with respect to their hypothesis spaces in consideration: the variational distribution q^* optimizes the ELBO over all the independent approximate distributions q, and the Gibbs posterior $Q^*(h)$ optimizes the PAC-Bayes bound (6) over all the base-learner Q.

3.3 PACTran with a Gamma Prior

Instead of using a set of l_1-normalized vectors $\mathbf{W}$, we can also relax the constraint by working on a matrix of non-negative variables $\mathbf{V} = \{v_{yz}\}$ whose prior is chosen to be the gamma distribution $P(v_{yz}) = Gamma(a_y, b)$. Unlike the normalized vectors $\mathbf{W}$, where $\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} M(\mathbf{x})_z w_{yz} = 1$ is automatically satisfied, when using unnormalized $\mathbf{V}$, we need to normalize the output explicitly,

$$p(y_i, z | \mathbf{x}_i, \mathbf{V}) = \frac{M(\mathbf{x}_i)_z v_{y_i z}}{\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} M(\mathbf{x}_i)_z v_{yz}}. \tag{11}$$

Note that $M(\mathbf{x}_i)$ is also not required to be normalized, which potentially makes the use case of Eq.(11) broader. Even with a normalized $M(\mathbf{x}_i)$, Eq.(11) strictly subsumes Eq.(8), because the former is only normalized once, while the latter is normalized $|\,\mathcal{Z}\,|$ times for each $\mathbf{w}_z$. In addition, since v_{yz} appears in both denominator and numerator, their scaling cancels out. Therefore, we fix a simple scaling coefficient $b = 1$ for all Gamma priors.

The rest of the Bayesian inference is similar to that of PACTran-Dirichlet. The PACTran-Gamma metric is the negative ELBO after applying the variational principles, and equals to (see details in A.2):

$$\begin{aligned}&\sum_y\sum_z (\log\Gamma(a_y) - \log\Gamma(\tilde{a}_{yz})) + \sum_i \log\tilde{\lambda}_i \\ +&\sum_i\sum_z q^*(z_i = z)\left(\log q^*(z_i = z) - \log M(x_i)_z\right),\end{aligned} \tag{12}$$

where,

$$\begin{aligned}&q^*(z_i = z) = \text{softmax}\left(\log M(\mathbf{x}_i)_z + \Psi(\tilde{a}_{y_i z})\right),\\ &\tilde{a}_{yz} = a_y + \sum_i q^*(z_i = z)\delta(y_i = y),\ \ \tilde{\lambda}_i = \sum_y\sum_z M(\mathbf{x}_i)_z\tilde{a}_{yz}.\end{aligned}$$

PACTran-Gamma metric is also a valid PAC-Bayesian upper bound to the generalization error, for the same reasons as the PACTran-Dirichlet metric.

3.4 PACTran with a Gaussian Prior

In the previous sections, we focus on the cases when the source model outputs normalized (in the Dirichlet prior case) or non-negative vectors (in the Gamma prior case). When the pretraining model is not based on classification tasks, one needs to add additional components (such as the Gaussian mixture models in $\mathcal{N}$-LEEP) to obtain those outputs. Here, we present another metric PACTran-Gaussian which relies only on penultimate layer representations $f(\mathbf{x})$. In PACTran-Gaussian, the prior P and posterior Q are both Gaussian distributions, where $P(\boldsymbol{\theta}) \sim \mathcal{N}(0, \sigma_0^2\mathbf{I})$ and $Q(\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\theta}_q, \mathbf{\Sigma}_q)$. For computational efficiency, we consider $\mathbf{\Sigma}_q = \sigma_q^2\mathbf{I}$ only. Note that although both LogME and PACTran-Gaussian use Gaussian priors and posteriors on $\boldsymbol{\theta}$, a main difference is that the former applies the squared loss, while the latter applies the cross-entropy loss (see more discussions in A.3). However, since the Gaussian prior is not conjugate to the exponentiated cross-entropy loss, we derive the bound using 2nd order approximations and a reparameterization trick as in [44],

$$\begin{aligned}
&\hat{L}(Q,S)+\frac{1}{\lambda}D_{KL}(Q\|P)\\
\simeq&\hat{L}(\boldsymbol{\theta}_q,S)+\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}[\sigma_q\boldsymbol{\epsilon}^\top\nabla\hat{L}(\boldsymbol{\theta}_q,S)+\frac{\sigma_q^2}{2}\boldsymbol{\epsilon}^\top\nabla^2\hat{L}(\boldsymbol{\theta}_q,S)\boldsymbol{\epsilon}\\
&\quad+\frac{1}{\lambda}\log\mathcal{N}(\boldsymbol{\theta}_q+\sigma_q\boldsymbol{\epsilon}|\boldsymbol{\theta}_q,\sigma_q^2\mathbf{I})-\frac{1}{\lambda}\log\mathcal{N}(\boldsymbol{\theta}_q+\sigma_q\boldsymbol{\epsilon}|0,\sigma_0^2\mathbf{I})]\\
=&\hat{L}(\boldsymbol{\theta}_q,S)+\frac{\sigma_q^2}{2}\mathrm{Tr}(\nabla^2\hat{L}(\boldsymbol{\theta}_q,S))+\frac{KD}{2\lambda}(\log\frac{\sigma_0^2}{\sigma_q^2}-1+\frac{\sigma_q^2}{\sigma_0^2}+\frac{\|\boldsymbol{\theta}_q\|_F^2}{KD\sigma_0^2}). \qquad (13)
\end{aligned}$$

The results of minimizing Eq. (13) with respect to σ_q and $\boldsymbol{\theta}_q$ yield the following optimal σ_* and $\boldsymbol{\theta}_*$ (see details in A.3),

$$\frac{\sigma_0^2}{\sigma_*^2}=1+\frac{\beta}{KD}\mathrm{Tr}(\nabla^2\hat{L}(\boldsymbol{\theta}_*,S)),\qquad \boldsymbol{\theta}_*=\operatorname*{argmin}_{\boldsymbol{\theta}_q}\left\{\hat{L}(\boldsymbol{\theta}_q,S)+\frac{\|\boldsymbol{\theta}_q\|_F^2}{2\beta}\right\},$$

where $\beta=\lambda\sigma_0^2$, So that we reach the following PACTran-Gaussian metric,

$$\underbrace{\hat{L}(\boldsymbol{\theta}_*,S)+\frac{\|\boldsymbol{\theta}_*\|_F^2}{2\beta}}_{RER}+\underbrace{\frac{KD\sigma_0^2}{2\beta}\log\frac{\sigma_0^2}{\sigma_*^2}}_{FR}. \qquad (14)$$

In Eq.(14), the first two terms are simply the l_2-regularized empirical risk (RER). The third term is a "flatness regularizer" (FR) that involves the trace of the Hessian of the empirical risk $\mathrm{Tr}(\nabla^2\hat{L}(\boldsymbol{\theta}_*,S))$ and has a simple closed-form solution for the cross-entropy loss (provided in A.3). It is accepted wisdom that a model generalizes better when its optimum is relatively flat [44, 14, 32] (low trace of Hessian). Empirically, we observe that the FR term is extremely effective in preventing the metrics from overfitting even though metric evaluation is done only on the training set.

It is also worth noting that there are two subtle, yet critical, differences between the derivations of our bound Eq.(14) and the ones in [44]. First, our mean parameter $\boldsymbol{\theta}_*$ is a minimum of Eq.(13), while in [44] it is an arbitrary model parameter. Second, in [44], σ_0 and σ_* were tied together during the optimization of σ_*, which violates the assumption of the PAC-Bayes theorem where the prior must be data independent. Instead, our σ_* is not only optimal, but also leaves σ_0 data independent.

4 Empirical Studies

In this section, we evaluate the PACTran metrics: PACTran-Dirichlet, PACTran-Gamma and PACTran-Gaussian, over several transfer learning benchmarks, and compare them against other existing transferability metrics including LEEP, NCE, $\mathcal{N}$-LEEP, H-Score, LogME.

4.1 The Neural Checkpoint Ranking Benchmark (NeuCRaB)

Pretrained Checkpoints Following NeuCRaB [29] (Group I), we collected a set of 16 ResNet-50 based checkpoints trained with various types of supervision. These checkpoints were pretrained on ImageNet with different training strategies, which include 5 models via self-supervised learning (Jigsaw [35], Relative Patch Location [12], Exemplar [13], Rotation [18], and Sup-Rotation [50]), 6 models via discriminators of generative models (WAE-UKL [39], WAE-GAN, WAE-MMD [41], Cond-BigGAN, Uncond-BigGAN [8], and VAE [26]), 2 via semi-supervised learning (Semi-Rotation-10% and Semi-Exemplar-10% [50]), one with a hybrid supervised loss (Sup-Exemplar-100% [50]), one by supervised learning of a standard Resnet50 (Sup-100% [21]), and lastly, one by supervised learning of a Resnet50 with identity mappings (Feature Vector [22]).

Downstream Tasks Following NeuCRaB [29], we adopt the Visual Task Adaptation Benchmark (VTAB) [51] and study diverse downstream tasks. The original NeuCRaB only contains four tasks: Caltech101 [15], Flowers102 [34], Patch Camelyon [46] and Sun397 [48]. In order to compare the transferability metrics on a wider variety of downstream tasks, we added 5 more tasks: DMLAB [51], CBIS-DDSM [40], Cifar10 [27], Oxford IIIT Pet [36] and Smallnorb(azimuth) [28]. These new tasks not only enrich the task categories, but also span the full range of the number of classes per tasks (single-digit, double-digit, and 100+ classes), which allows us to analyze the performance of transferability metrics according to the number of classes. In particular, we group these tasks according to the number of output classes: tasks with 100+ classes include Caltech101 (102 classes), Flowers102 (102 classes), Sun397 (397 classes); tasks with 10-99 classes include Cifar10 (10 classes), Oxford IIIT Pet (37 classes) and Smallnorb(azimuth) (18 classes); tasks with 2-9 classes include Patch Camelyon (2 classes), DMLAB (6 classes), and CBIS-DDSM (5 classes).

Evaluating the Transferability Metrics We use the Kendall-Tau rank correlation coefficient to correlate between the transferability metric scores and the testing error of the finetuned checkpoints. The "ground-truth" testing error that corresponds to each pretrained checkpoint M is obtained by finetuning M on the downstream training set multiple times and setting the ground-truth testing error e_M to the lowest test error among the runs (See details in B.3).

Experimental Settings Since it is crucial for a transferability metric to be highly efficient compared to the finetuning, we focus our experiments on limited-data settings. Let K denote the number of classes, D the feature dimension and N the number of examples for computing the metric. We consider three data settings with increasing average number of samples per class $N/K \in \{2, 5, 10\}$ (to avoid having too few examples, we also set a lower bound for $N \geq 20$). For each N/K setting, we subsample N samples from the training set of each downstream task 5 times. The transferability metrics are then evaluated over the 5 splits and their average Kendall-Tau correlation is reported. Compared to evaluating the metrics on the full training set, the limited-data setting significantly reduces the cost of penultimate-layer feature extraction, which is usually orders of magnitude more expensive than computing the metrics themselves (see Table. 2).

Besides the aforementioned baseline transferability metrics (LEEP, $\mathcal{N}$-LEEP, H-score, LogME), we also include two additional metrics based on linear classification. The LINEAR metric is based on the training loss of a regularized linear classifier (the sum of the first two terms of Eq.(14)). The second metric LINEAR-VALID splits the subsampled dataset into two equally sized folds, trains a linear classifer on one fold and computes the validation error on the other. The regularizing coefficients for both metrics are $\beta \in \{0.1, 1.0, 10\} \cdot N$. For LINEAR-VALID, the model with the lowest validation error is chosen. For LINEAR, since there is no validation set, we select the β that maximizes the Kendall correlation between the loss and LINEAR-VALID's validation error across checkpoints.

For $\mathcal{N}$-LEEP, we follow the recipe from [29] and set the PCA energy percentage to 80% and the number of Gaussian components to the number of classes in the downstream task. For PACTran-Dirichlet and PACTran-Gamma, we set $\alpha_y = \hat{p}(y)$. For PACTran-Gaussian, we report two sets of results: In PT-Gauss$_{fix}$, we fix the two hyperparameters to $\beta = 10N$ and $\sigma_0^2 = \frac{100}{D}$. In PT-Gauss$_{grid}$, we perform a hyperparameters grid-search over $\beta \in \{0.1, 1.0, 10\} \cdot N$ and $\sigma_0^2 \in \{1.0, 10, 100, 1000\} \cdot \frac{1}{D}$ and select the hyperparameters (β, σ_0^2) that maximize the Kendall correlation between the PT-Gauss$_{grid}$ metric scores and LINEAR-VALID's validation errors across checkpoints. More detailed discussions about the hyperparameters are available in B.5.

Results and Analysis Table 1 reports the Kendall-Tau correlations of the various transferability metrics. Table 2 reports the GFLOPS per metric for each task as well as those of the feature extraction stage from the pretrained checkpoints. For LINEAR and PT-Gauss$_{grid}$, we include the GFLOPS for all hyperparameter runs as well as hyperparameter selection for LINEAR-VALID.

Although LEEP is the fastest algorithm, its averaged performance is worse than most other metrics. All other metrics employ more expensive components (PCA and GMM for $\mathcal{N}$-LEEP and $\mathcal{N}$-PT-Dir/Gam, SVD for Hscore and LogME, and L-BFGS for LINEAR and PT-Gauss) but are still 1-2 orders of magnitude faster to compute than feature extraction from the penultimate layer.

$\mathcal{N}$-LEEP, which obtains the source class assignments by applying GMM on the penultimate layer outputs, performs much better than LEEP on average. In addition, PACTran-Dirichlet and PACTran-Gamma with the GMM assignments (denoted as $\mathcal{N}$-PT-Dir and $\mathcal{N}$-PT-Gam) perform similarly to the $\mathcal{N}$-LEEP algorithm, which indicates that the EEP estimator is surprisingly close to Bayesian optimal based on the GMM assignments of the VTAB tasks.

Among the algorithms that use the L-BFGS optimizer, LINEAR-VALID performs better than LINEAR for large K. However, for small K LINEAR-VALID becomes worse, probably because the training and validation splits are too small. In contrast, the PT-Gauss metrics are consistently among the best metrics across all settings, which provides clear evidence that the 3rd "flatness" term (Eq.(14)) plays a crucial role in predicting generalization error. For example, they are the only metrics with correlation 0.4 or higher on 2-9 classes.

In comparing between PT-Gauss$_{grid}$ and PT-Gauss$_{fix}$, we find that PT-Gauss$_{grid}$ usually performs well whenever LINEAR-VALID's does (since it de-

$N/K = 2$	100+ classes	10-99 classes	2-9 classes	Average
LEEP	0.202	0.005	0.041	0.083
$\mathcal{N}$-LEEP	0.723	0.401	0.077	0.401
H-score	0.413	0.106	0.185	0.235
LogME	0.308	0.067	0.071	0.149
LINEAR	0.231	0.072	0.114	0.139
LINEAR-VALID	0.750	0.309	0.063	0.374
$\mathcal{N}$-PT-Dir	0.760	0.327	0.099	0.395
$\mathcal{N}$-PT-Gam	0.763	0.333	0.108	0.401
PT-Gauss$_{grid}$	**0.868**	0.664	**0.509**	**0.680**
PT-Gauss$_{fix}$	0.770	**0.683**	**0.509**	0.654
$N/K = 5$	100+ classes	10-99 classes	2-9 classes	Average
LEEP	0.112	0.082	0.023	0.109
$\mathcal{N}$-LEEP	0.795	0.536	0.096	0.476
H-score	0.412	0.141	0.118	0.224
LogME	0.421	0.093	0.075	0.196
LINEAR	0.253	0.084	0.122	0.153
LINEAR-VALID	0.807	0.411	0.044	0.420
$\mathcal{N}$-PT-Dir	**0.826**	0.458	0.140	0.475
$\mathcal{N}$-PT-Gam	**0.825**	0.462	0.151	0.479
PT-Gauss$_{grid}$	0.793	**0.716**	0.412	0.641
PT-Gauss$_{fix}$	**0.832**	0.675	**0.512**	**0.673**
$N/K = 10$	100+ classes	10-99 classes	2-9 classes	Average
LEEP	0.276	0.079	0.049	0.134
$\mathcal{N}$-LEEP	0.822	0.520	0.148	0.497
Hscore	0.461	0.318	0.158	0.313
LogME	0.488	0.138	0.073	0.233
LINEAR	0.325	0.089	0.109	0.174
LINEAR-VALID	**0.835**	0.482	0.123	0.480
$\mathcal{N}$-PT-Dir	**0.839**	0.446	0.134	0.473
$\mathcal{N}$-PT-Gam	**0.839**	0.452	0.140	0.477
PT-Gauss$_{grid}$	0.769	**0.678**	0.429	0.625
PT-Gauss$_{fix}$	0.778	0.609	**0.534**	**0.641**

Table 1. Kendall-Tau correlations on the NeuCRaB experiments with different N/K, where K is the number of classes, N the number of examples for computing the metric.

pends on it for hyperparameter selection). On the other hand, when LINEAR-VALID is worse (K is small), PT-Gauss$_{fix}$ outperforms PT-Gauss$_{grid}$.

4.2 Visual Question Answering

We further conduct experiments over the multi-modal VQA task. Following common practice ([2, 20]), we treat VQA as a classification task (vocab-based VQA). That is, we construct a vocabulary based on the top answers in the training sets and classify into one of those labels.

GFLOPS	100+ classes	10-99 classes	2-9 classes
LEEP	6.40E-1	2.00E-2	1.55E-3
$\mathcal{N}$-LEEP	2.09E2	1.40E0	7.25E-2
Hscore	1.33E2	1.30E2	1.29E2
LogME	1.34E2	1.30E2	1.29E2
LINEAR	2.89E2	9.03E0	7.07E-1
LINEAR-VALID	9.64E1	3.02E0	2.36E-1
$\mathcal{N}$-PT-Dir	2.09E2	1.40E0	7.25E-2
$\mathcal{N}$-PT-Gam	2.09E2	1.40E0	7.25E-2
PT-Gauss$_{grid}$	2.90E2	9.07E0	7.10E-1
PT-Gauss$_{fix}$	6.45E1	2.02E0	1.58E-1
Penultimate Feature	3.88E3	6.84E2	1.90E2

Table 2. GFLOPS of running each metrics and the penultimate-layer feature extraction stage on the subsampled datasets, when $N/K = 10$.

Pretrained Checkpoints We apply the state-of-art VQA model architecture, which fuses image and question representations in a multimodal Transformer model [45] (see C.1). We pretrain the VQA models over 9 datasets, including: VQA-v2 [20], GQA [24], V7W [53], CNETVQA, TP-COLOR-COCO, TP-COLOR-CC3M, TP-COLOR-CC12M, VQ2A-COCO, and VQ2A-CC3M [10]. The detailed descriptions of the datasets are provided in C.2.

For each pretraining dataset, we consider 3 different model sizes and 4 different finetuning hyperparameter settings. For model sizes, the number of layers t of the text-encoder and m of the multimodal-encoder is varied from $(t, m) \in \{(6, 3), (9, 5), (12, 7)\}$. For hyperparameters, the dropout ratios are varied from $\{0, 0.1\}$. We use two learning schedules: a constant learning rate of 0.0005 and a decay learning rate starting at 0.2. For each of these 12 settings, we set batch size to 128, and save a checkpoint after 100,000 iterations.

Downstream Task We chose the OKVQA dataset [30] because the task requires additional knowledge beyond its own training set, and it has been shown that proper pretraining brings significant benefits to performance [30, 10].

Experimental Settings Finetuning details are available in C.3. The hyperparameter settings match the NeuCRaB experiments. Otherwise, we vary the number of data examples for metric computation from $N \in \{40, 100, 200\}$ and restrict the examples from the top 20 answers such that $N/K \in \{2, 5, 10\}$. For each N, we create 5 subsamples of the OKVQA train set. Each metric is then evaluated on the 5 splits and the average correlation score is reported.

Results and Analysis In total, there are $108 = 9 \times 12$ checkpoints that span 9 different pretraining datasets, and 12 different model configurations. In Table 3, we report their results in 3 different ways: (1) "CD" (cross pretraining data sources), reports the averaged correlation of metrics across the 9 different pretraining datasets for each of the 12 model configurations; (2) "CM" (cross models), reports the averaged correlation of metrics cross the 12 model configurations for each of the 9 pretraining datasets; and (3) "Total", reports the correlation over all 108 checkpoints.

N	40			100			200		
	CD	CM	Total	CD	CM	Total	CD	CM	Total
LEEP	0.420	0.337	0.471	0.430	0.373	0.492	0.435	0.402	0.508
$\mathcal{N}$-LEEP	0.309	0.077	0.295	0.452	0.232	0.427	0.503	0.329	0.480
Hscore	0.220	0.048	0.198	0.253	0.079	0.222	0.233	0.116	0.243
LogME	0.350	0.141	0.402	0.343	0.154	0.395	0.357	0.160	0.397
LINEAR	0.355	0.137	0.410	0.351	0.167	0.407	0.382	0.209	0.423
LINEAR-VALID	**0.488**	0.118	0.430	0.526	0.172	0.474	0.579	0.360	0.528
PT-Dir	0.253	0.329	0.301	0.449	**0.418**	0.480	0.460	**0.469**	0.503
PT-Gam	0.453	**0.348**	**0.490**	0.518	**0.411**	**0.544**	0.522	0.430	0.532
$\mathcal{N}$-PT-Dir	0.424	0.093	0.358	0.522	0.277	0.476	0.548	0.335	0.504
$\mathcal{N}$-PT-Gam	0.421	0.092	0.353	0.524	0.278	0.474	0.547	0.333	0.504
PT-Gauss$_{grid}$	**0.480**	0.272	0.451	**0.566**	0.349	**0.544**	**0.617**	0.391	**0.582**

Table 3. Kendall-Tau correlations on the OKVQA experiments with different N.

As can be seen, when the pretraining tasks are classification based, LEEP performs much better compared to the "mixed supervision" tasks in the previous section. On the other hand, PACTran-Gamma outperforms LEEP and PACTran-Dirichlet, which indicates that an unnormalized weight transfer matrix is more helpful in these setting. LINEAR-VALID is a strong baseline, especially as more data examples are provided. Finally, we see that PACTran-Gauss (with $\beta = 0.1N$ and $\sigma_0^2 = \frac{1}{D}$ from the grid search) provides competitive performance in all cases, and is consistently among the best in evaluating transferability from different pretraining datasets ("CD").

5 Conclusion

In this paper we presented PACTran, a PAC-Bayesian based framework for measuring the transferability of pretrained checkpoints to downstream tasks. Our method significantly improves upon previous methods in that it is both theoretically sound as well as compatible with downstream classification tasks. We instantiated three variant PACTran metrics using different hypothesis spaces and priors and conducted experiments over a set of vision tasks (VTAB) and a vision-and-language task (OKVQA). We showed that some PACTran variants can provide theoretical justification for existing methods. For example, ($\mathcal{N}$-)PT-Dir and ($\mathcal{N}$-)PT-Gam metrics subsume ($\mathcal{N}$-)LEEP, in which the finetuning head sits on top of the pretrained classification head (or a GMM). Our experiments also showed that several of the baseline metrics are unable to measure checkpoint transferability better than a simple linear classification and validation baseline (LINEAR-VALID). On the other hand, the proposed PT-Gauss metric behaved well as a measure of transferability in a limited data setting and consistently exhibited high correlation with the test performance of models finetuned on the downstream tasks. Possibly, this is because it more closely matches the setup of the finetuned model, where the finetuning head is placed directly on the penultimate layer and trained with a cross-entropy loss.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: VQA: Visual question answering. In: ICCV (2015)
3. Bao, Y., Li, Y., Huang, S.L., Zhang, L., Zheng, L., Zamir, A., Guibas, L.: An information-theoretic approach to transferability in task transfer learning. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 2309–2313. IEEE (2019)
4. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American statistical Association **112**(518), 859–877 (2017)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. the Journal of machine Learning research **3**, 993–1022 (2003)
6. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramèr, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., Liang, P.: On the opportunities and risks of foundation models (2021)
7. Bousquet, O., Boucheron, S., Lugosi, G.: Introduction to statistical learning theory. In: Summer school on machine learning. pp. 169–207. Springer (2003)
8. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis (2019)
9. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
10. Changpinyo, S., Kukliansky, D., Szpektor, I., Chen, X., Ding, N., Soricut, R.: All you may need for vqa are image captions. In: NAACL (2022)
11. Ding, N., Chen, X., Levinboim, T., Goodman, S., Soricut, R.: Bridging the gap between practice and pac-bayes theory in few-shot meta-learning. Advances in Neural Information Processing Systems **34** (2021)

12. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction (2016)
13. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014)
14. Dziugaite, G.K., Roy, D.M.: Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008 (2017)
15. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Pattern Recognition Workshop (2004)
16. Germain, P., Bach, F., Lacoste, A., Lacoste-Julien, S.: PAC-bayesian theory meets bayesian inference. Advances in Neural Information Processing Systems **29**, 1884–1892 (2016)
17. Germain, P., Lacasse, A., Laviolette, F., Marchand, M.: PAC-bayesian learning of linear classifiers. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 353–360 (2009)
18. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations (2018)
19. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
20. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
22. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks (2016)
23. Huang, S.L., Makur, A., Wornell, G.W., Zheng, L.: On universal features for high-dimensional learning and inference. arXiv preprint arXiv:1911.09105 (2019)
24. Hudson, D.A., Manning, C.D.: GQA: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR (2019)
25. Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., Bengio, S.: Fantastic generalization measures and where to find them. In: ICLR (2020)
26. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2014)
27. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
28. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2**, II–104 Vol.2 (2004)
29. Li, Y., Jia, X., Sang, R., Zhu, Y., Green, B., Wang, L., Gong, B.: Ranking neural checkpoints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2663–2673 (2021)
30. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

31. McAllester, D.A.: Some PAC-bayesian theorems. Machine Learning **37**(3), 355–363 (1999)
32. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. Advances in neural information processing systems **30** (2017)
33. Nguyen, C., Hassner, T., Seeger, M., Archambeau, C.: Leep: A new measure to evaluate transferability of learned representations. In: International Conference on Machine Learning. pp. 7294–7305. PMLR (2020)
34. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (Dec 2008)
35. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles (2017)
36. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
37. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR (2020)
38. Rothfuss, J., Fortuin, V., Josifoski, M., Krause, A.: Pacoh: Bayes-optimal meta-learning with pac-guarantees. In: International Conference on Machine Learning. pp. 9116–9126. PMLR (2021)
39. Rubenstein, P., Bousquet, O., Djolonga, J., Riquelme, C., Tolstikhin, I.O.: Practical and consistent estimation of f-divergences. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
40. Sawyer-Lee, R., Gimenez, F., Hoogi, A., Rubin, D.: Curated breast imaging subset of ddsm (2016). https://doi.org/10.7937/k9/tcia.2016.7o02s9cy
41. Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein auto-encoders (2019)
42. Tran, A.T., Nguyen, C.V., Hassner, T.: Transferability and hardness of supervised classification tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1395–1405 (2019)
43. Tripuraneni, N., Jordan, M., Jin, C.: On the theory of transfer learning: The importance of task diversity. Advances in Neural Information Processing Systems **33**, 7852–7862 (2020)
44. Tsuzuku, Y., Sato, I., Sugiyama, M.: Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using PAC-Bayesian analysis. In: Proceedings of the 37th International Conference on Machine Learning. pp. 9636–9647 (2020)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
46. Veeling, B.S., Linmans, J., Winkens, J., Cohen, T., Welling, M.: Rotation equivariant cnns for digital pathology (Sep 2018). https://doi.org/10.1007/978-3-030-00934-2-24
47. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771 (2019)
48. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3485–3492 (June 2010). https://doi.org/10.1109/CVPR.2010.5539970

49. You, K., Liu, Y., Wang, J., Long, M.: Logme: Practical assessment of pre-trained models for transfer learning. In: International Conference on Machine Learning. pp. 12133–12143. PMLR (2021)
50. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1476–1485 (2019). https://doi.org/10.1109/ICCV.2019.00156
51. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., et al.: A large-scale study of representation learning with the visual task adaptation benchmark. arXiv preprint arXiv:1910.04867 (2019)
52. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)
53. Zhu, Y., Groth, O., Bernstein, M., Li, F.F.: Visual7W: Grounded question answering in images. In: CVPR (2016)