# Newer is not always better: Rethinking transferability metrics, their peculiarities, stability and performance

Shibal Ibrahim[1⋆](✉), Natalia Ponomareva[2], and Rahul Mazumder[1]

[1] Massachusetts Institute of Technology, Cambridge MA, USA
{shibal,rahulmaz}@mit.edu
[2] Google Research, New York NY, USA
nponomareva@google.com

**Abstract.** Fine-tuning of large pre-trained image and language models on small customized datasets has become increasingly popular for improved prediction and efficient use of limited resources. Fine-tuning requires identification of best models to transfer-learn from and quantifying transferability prevents expensive re-training on *all* of the candidate models/tasks pairs. In this paper, we show that the statistical problems with covariance estimation drive the poor performance of H-score — a common baseline for newer metrics — and propose shrinkage-based estimator. This results in up to 80% absolute gain in H-score correlation performance, making it competitive with the state-of-the-art LogME measure. Our shrinkage-based H-score is $3-10$ times faster to compute compared to LogME. Additionally, we look into a less common setting of target (as opposed to source) task selection. We demonstrate previously overlooked problems in such settings with different number of labels, class-imbalance ratios etc. for some recent metrics e.g., NCE, LEEP that resulted in them being misrepresented as leading measures. We propose a correction and recommend measuring correlation performance against relative accuracy in such settings. We support our findings with $\sim$ 164,000 (fine-tuning trials) experiments on both vision models and graph neural networks.

**Keywords:** Transferability metrics · Fine-tuning · Transfer learning · Domain adaptation · Neural networks.

## 1 Introduction

Transfer learning is a set of techniques of using abundant somewhat related source data $p(X^{(s)}, Y^{(s)})$ to ensure that a model can generalize well to the target domain, defined as either little amount of labelled data $p(X^{(t)}, Y^{(t)})$ (supervised), and/or a lot of unlabelled data $p(X^{(t)})$ (unsupervised transfer learning). Transfer learning is most commonly achieved either via fine-tuning or co-training. Fine-tuning is a process of adapting a model trained on source data by using target

⋆ This work was completed as an Intern and Student Researcher at Google.

data to do several optimization steps (for example, stochastic gradient descent) that update the model parameters. Co-training on source and target data usually involves reweighting the instances in some way or enforcing domain irrelevance on feature representation layer, such that the model trained on such combined data works well on target data. Fine-tuning is becoming increasing popular because large models like ResNet50 [11], BERT [6] etc. are released by companies and are easily modifiable. Training such large models from scratch is often prohibitively expensive for the end user.

In this paper, we are primarily interested in effectively measuring transferability before training of the final model begins. Given a source data/model, a **transferability measure** quantifies how much knowledge of source domain/model is transferable to the target model. Transferability measures are important for various reasons: they allow understanding of relationships between different learning tasks, selection of highly transferable tasks for joint training on source and target domains, selection of optimal pre-trained source models for the relevant target task, prevention of trial procedures attempting to transfer from each source domain and optimal policy learning in reinforcement learning scenarios (e.g. optimal selection of next task to learn by a robot). If a measure is capable of efficiently and accurately measuring transferability across arbitrary tasks, the problem of task transfer learning is greatly simplified by using the measure to search over candidate transfer sources and targets.

***Contributions*** Our contributions are three-fold:

1. We show that H-score, commonly used as a baseline for newer transferability measures, suffers from instability due to poor estimation of covariance matrices. We propose shrinkage-based estimation of H-score with regularized covariance estimation techniques from statistical literature. We show 80% absolute increase over the original H-score and show superior performance in majority cases against all newer transferability measures across various fine-tuning scenarios.
2. We present a fast implementation of our estimator that is $3-10$ times faster than state-of-the-art LogME measure.
3. We identify problems with 3 other transferability measures (NCE, LEEP and $\mathcal{N}$LEEP) in target task selection when either the number of target classes or the class imbalance varies across candidate target tasks. We propose measuring correlation against relative target accuracy (instead of vanilla accuracy) in such scenarios.

Our large set of $\sim 164,000$ fine-tuning experiments with vision models and graph convolutional networks on real-world datasets shows usefulness of our proposals.

This paper is organized as follows. Section 2 describes general fine-tuning regimes and transfer learning tasks. Section 3 discusses transferability measures. Section 4 addresses shortcomings of the pioneer transferability measure (H-Score) that arise due to unreliable estimation and proposes a new shrinkage-based estimator for the H-Score. In Section 5, we demonstrate problems with recent NCE, LEEP and $\mathcal{N}$LEEP metrics and propose a way to address them. Finally, Section 6 presents a meta study of all metrics.

## 2 Transferability setup

We consider the following fine-tuning scenarios based on existing literature.
(i) *Source Model Selection (SMS)*: For a particular target data/task, this regime aims to select the "optimal" source model (or data) to transfer-learn from, from a collection of candidate models/data.
(ii) *Target Task Selection (TTS)*: For a particular (source) model, this regime aims to find the most related target data/task.

In addition, we primarily consider two different fine-tuning strategies:
(i) *Linear fine-tuning/head only fine-tuning (LFT)*: All layers except for the penultimate layer are frozen. Only the weights of the head classifier are re-trained while fine-tuning.
(ii) *Non-linear fine-tuning (NLFT)*: Any layer can be designated as a feature extractor, up to which all the layers are frozen; the subsequent layers include nonlinear transformations and are re-trained along with the head on target data.

## 3 Related Work

Recent literature in transfer learning has proposed efficient transferability measures. Inspired by principles in information theory, Negative Conditional Entropy (NCE) [31] uses pre-trained source model and evaluates conditional entropy between target pseudo labels (source models' assigned labels) and real target labels. Log Expected Empirical Predictor (LEEP) [21] modifies NCE by using soft predictions from the source model. Both NCE and LEEP do not directly use feature information, hence they are not applicable for layer selection. The authors in [4] propose representing each output class by the mean of all images from that class and computing Earth Mover's distance between the centroids of the source classes and target classes.

Other works [1,18,12,33,5] proposed metrics that capture information from both the (learnt) feature representations and the real target labels. These metrics are more appealing as these can be broadly applicable for models that are pre-trained in either supervised or unsupervised fashion. They are also applicable for embedding layer selection. The authors in [18] proposed $\mathcal{N}$LEEP that fits a Gaussian mixture model on the target feature embeddings and computes the LEEP score between the probabalistic assignment of target features to different clusters and the target labels. The authors in [12] introduced TransRate — a computationally-friendly surrogate of mutual information (using coding rate) between the target feature embeddings and the target labels. H-score was proposed by [1] that takes into account inter-class feature variance and feature redundancy. [33] proposed LogME that considers an optimization problem rooted in Bayesian statistics to maximize the marginal likelihood under a linear classifier head. [5] introduced LFC to measure in-class similarity of target feature embeddings across samples.

Finally, the authors in [30] used Optimal Transport to evaluate domain distance, and combined it, via a linear combination, with NCE. To learn such a

measure, a portion of target tasks were set aside, the models were transferred onto these tasks and the results were used to learn the coefficients for the combined Optimal Transport based Conditional Entropy (OTCE) metric. While the resulting metric appears to be superior over other non-composite metrics like H-score and LEEP, it is expensive to compute since it requires finding the appropriate coefficients for the combination.

## 4 Improved estimation of H-score

*H-score* [1] is one of the pioneer measures that is often used as a baseline for newer transferability measures, which often demonstrate the improved performance. It characterizes discriminatory strength of feature embedding for classification:

$$\mathrm{H}(f) = \mathrm{tr}(\boldsymbol{\Sigma}^{f^{-1}}\boldsymbol{\Sigma}^{z}) \tag{1}$$

where, $d$ is the embedding dimension, $\boldsymbol{f}_i = h(\boldsymbol{x}_i^{(t)}) \in \mathbb{R}^d$ is the target feature embeddings when the feature extractor $(h : \mathbb{R}^p \to \mathbb{R}^d)$ from the source model is applied to the target sample $\boldsymbol{x}_i^{(t)} \in \mathbb{R}^p$, $\boldsymbol{F} \in \mathbb{R}^{n_t \times d}$ denotes the corresponding target feature matrix, $Y = Y^{(t)} \in \mathcal{Y} = \{1, \cdots, C\}$ are the target data labels, $\boldsymbol{\Sigma}^f \in \mathbb{R}^{d\times d}$ denotes the sample feature covariance matrix of $\boldsymbol{f}$, $\boldsymbol{z} = \mathbb{E}\left[\boldsymbol{f}|Y\right] \in \mathbb{R}^d$ and $\boldsymbol{Z} \in \mathbb{R}^{n_t \times d}$ denotes the corresponding target-conditioned feature matrix, $\boldsymbol{\Sigma}^z \in \mathbb{R}^{d\times d}$ denotes the sample covariance matrix of $\boldsymbol{z}$. Intuitively, $\mathrm{H}(f)$ captures the notion that higher inter-class variance and small feature redundancy results in better transferability.

We hypothesize that the sub-optimal performance of H-Score (compared to that of more recent metrics) for measuring transferability in many of the evaluation cases, e.g., in [21], is due to lack of robust estimation of H-Score. We empirically validate this hypothesis using a synthetic classification data. We generated 1 million 1000-dimensional features with 10 classes using Sklearn multi-class dataset generation function [22]. Number of informative features is set to 500 with rest filled with random noise. We visualize the original and the population version of the H-score for different sample sizes in Fig. 1. We observe that the original H-Score becomes highly unreliable as the number of samples decreases.

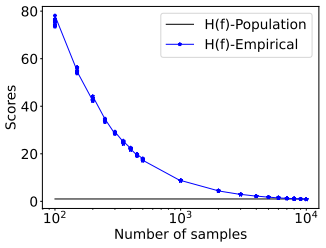


Fig. 1: Non-reliability of $\mathrm{H}(f)$. $\mathrm{H}(f)$ is $\sim 75\times$ larger than the population version of the H-Score. Population version is estimated with $10^6$ samples.

Many of the deep learning models in the context of transfer learning have high-dimensional feature embedding space — typically larger than the number of target samples. Consequently, the estimation of the two covariance matrices in H-score becomes challenging: the sample covariance matrix of the feature embedding has a large

condition number[3] in small data regimes. In many cases, it cannot even be inverted. [1] used a pseudo-inverse of the covariance matrix $\boldsymbol{\Sigma}^f$. However, this method of estimating a precision matrix can be sub-optimal as inversion can amplify estimation error [16]. We propose to use well-conditioned shrinkage estimators motivated by the rich literature in statistics on the estimation of high-dimensional covariance (and precision) matrices [23]. We show that the use of such shrinkage estimators can offer significant gain in the performance of H-score in predicting transferability. In many cases, as our experiments show, the gain is so significant that H-score becomes a leading transferability measure, surpassing the performance of state-of-the-art measures.

### 4.1 Proposed Transferability Measure

We propose the following shrinkage based H-score:

$$\mathrm{H}_\alpha(f) = \mathrm{tr}\big(\boldsymbol{\Sigma}_\alpha^{f^{-1}} \cdot (1-\alpha)\boldsymbol{\Sigma}^z\big), \tag{2}$$

***Estimating*** $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}^{(\boldsymbol{f})}$ While there are several possibilities to obtain a regularized covariance matrix [23], we present an approach that considers a linear operation on the eigenvalues of the sample version of the feature embedding covariance matrix. Similar ideas of using well-conditioned plug-in covariance matrices are used in the context of discriminant analysis [10]. In particular, we improve the conditioning of the covariance matrix by considering its weighted convex combination with a scalar multiple of the identity matrix:

$$\boldsymbol{\Sigma}_\alpha^f = (1-\alpha)\boldsymbol{\Sigma}^f + \alpha\sigma\boldsymbol{I}_d \tag{3}$$

where $\alpha \in [0,1]$ is the shrinkage parameter and $\sigma$ is the average variance computed as $\mathrm{tr}(\boldsymbol{\Sigma}^f)/d$. The linear operation on the eigenvalues ensures the covariance estimator is positive definite. Note that the inverse of $\boldsymbol{\Sigma}_\alpha^f$ can be computed for every $\alpha$, by using the eigen-decomposition of $\boldsymbol{\Sigma}^f$. The shrinkage parameter controls the bias and variance trade-off; the optimal $\alpha$ needs to be selected. This distribution-free estimator is well-suited for our application as the explicit convex linear combination is easy to compute and makes the covariance estimates well-conditioned and more accurate [16,2,28].

***Understanding*** $(\mathbf{1}-\boldsymbol{\alpha})\boldsymbol{\Sigma}^{\boldsymbol{z}}$ The scaling factor $(1-\alpha)$ can be understood in terms of regularized covariance matrix estimation under a ridge penalty:

$$1/(1+\lambda)\cdot\boldsymbol{\Sigma}^z = \mathrm{argmin}_{\hat{\boldsymbol{\Sigma}}}\,||\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}^z||_2^2 + \lambda||\hat{\boldsymbol{\Sigma}}||_2^2 \tag{4}$$

where $\lambda \geq 0$ is the ridge penalty. Choosing $\lambda = \alpha/(1-\alpha)$, it becomes clear that $(1-\alpha)\boldsymbol{\Sigma}^z$ is the regularized covariance matrix.

***Choice of*** $\boldsymbol{\alpha}$ [16] proposed a covariance matrix estimator that minimizes mean squared error loss between the shrinkage based covariance estimator and

[3] Condition number of a positive semidefinite matrix $A$, is the ratio of its largest and smallest eigenvalues.

the true covariance matrix. The optimization considers the following objective:

$$\min_{\alpha,v} \ \mathbb{E}[||\boldsymbol{\Sigma}^* - \boldsymbol{\Sigma}||^2] \qquad \text{s.t.} \ \ \boldsymbol{\Sigma}^* = (1-\alpha)\boldsymbol{\Sigma}^f + \alpha v I, \quad \mathbb{E}[\boldsymbol{\Sigma}^f] = \boldsymbol{\Sigma}. \tag{5}$$

where $\|\boldsymbol{A}\|^2 \stackrel{\text{def}}{=} \text{tr}(\boldsymbol{A}\boldsymbol{A}^T)/d$ for a matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$. This optimization problem permits a closed-form solution for the optimal shrinkage parameter, which is given by:

$$\alpha^* = \mathbb{E}[||\boldsymbol{\Sigma}^f - \boldsymbol{\Sigma}||^2]/\mathbb{E}[||\boldsymbol{\Sigma}^f - (\text{tr}(\boldsymbol{\Sigma})/d)\cdot \boldsymbol{I}_d||^2] \tag{6}$$

$$\simeq \min\{(1/n_t^2)\sum_{i\in[n_t]} \frac{||\boldsymbol{f}_i\boldsymbol{f}_i^T - \boldsymbol{\Sigma}^f||^2}{||\boldsymbol{\Sigma}^f - (\text{tr}(\boldsymbol{\Sigma}^f)/d)\cdot \boldsymbol{I}_d||^2}, 1\}. \tag{7}$$

where equation 7 defines a valid estimator (not dependent on true covariance matrix) for practical use. For proof, we refer the readers to Section 2.1 and 3.3 in [16].

Following the synthetic motivational example presented earlier showing the unreliability of the original H-Score, we investigate the reliability of shrinkage-based H-Score. We visualize the shrinkage-based H-Score in Fig. 2[Left]. We observe that the original H-Score becomes highly unreliable as the number of samples decreases. In contrast, the shrunken estimation of H-Score is highly stable and has a small error when compared with the population H-Score. Hence, shrinkage-based H-score seems to be a much better estimator of the "true" H-score in contrast to the empirical H-Score. We further visualize the effect of using non-optimal values of $\alpha$ on the shrinkage-based H-Score in Fig. 2[Right]. We can see that the shrinkage-based H-Score with optimal shrinkage $\alpha^*$ is much closer to the population version of the original H-Score, especially for smaller sample cases. This validates the use of $\alpha^*$ as computed in equation 7.

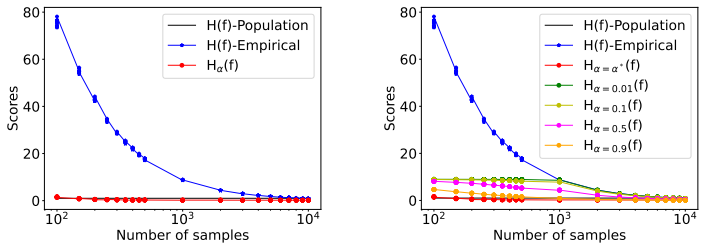


Fig. 2: [Left] Stability of $\text{H}(f)$ and our shrinkage-based $\text{H}_\alpha(f)$ with respect to number of samples. $\text{H}(f)$ is $\sim 75$ times larger than the population version of the H-Score (estimated with a sample size of $10^6$). In contrast, the shrinkage-based H-Score is significantly more reliable. [Right] Effect of $\alpha$ on shrunk H-Score.

***Additional Discussion on same shrinkage $\alpha$ for the two covariances in shrinkage-based H-Score*** The covariance $\boldsymbol{\Sigma}^z$ can not be shrunk independently of $\boldsymbol{\Sigma}^f$ in the estimation of $\text{H}_\alpha(f)$— the two covariances are coupled by

the law of total covariance:

$$\boldsymbol{\Sigma}^f = \mathbb{E}[\boldsymbol{\Sigma}^{f_Y}] + \boldsymbol{\Sigma}^z. \tag{8}$$

where $\boldsymbol{f}_Y$ denotes the feature embedding of target samples belonging to class $Y \in \mathcal{Y}$ and $\boldsymbol{\Sigma}^{f_Y} = \mathrm{Cov}(\boldsymbol{f}|Y)$ denotes the class-conditioned covariances. We write

$$(1-\alpha)\boldsymbol{\Sigma}^f = (1-\alpha)\mathbb{E}[\boldsymbol{\Sigma}^{f_Y}] + (1-\alpha)\boldsymbol{\Sigma}^z,$$

$$(1-\alpha)\boldsymbol{\Sigma}^f + \alpha\frac{\mathrm{tr}(\boldsymbol{\Sigma}^f)}{d}\boldsymbol{I}_d = (1-\alpha)\mathbb{E}[\boldsymbol{\Sigma}^{f_Y}] + \alpha\frac{\mathrm{tr}(\boldsymbol{\Sigma}^f)}{d}\boldsymbol{I}_d + (1-\alpha)\boldsymbol{\Sigma}^z, \tag{9}$$

$$\text{i.e, } \boldsymbol{\Sigma}^f_\alpha = (1-\alpha)\mathbb{E}[\boldsymbol{\Sigma}^{f_Y}] + \alpha\frac{\mathrm{tr}(\boldsymbol{\Sigma}^f)}{d}\boldsymbol{I}_d + (1-\alpha)\boldsymbol{\Sigma}^z. \tag{10}$$

Comparing equations 10 with 8, we see that the same shrinkage parameter $\alpha$ should be used when using shrinkage estimators, to preserve law of total covariance. The first two terms on the right side in equation 10 can be understood as shrinkage of class-conditioned covariances to the average (global) variance. The third term in equation 10 (e.g. $(1-\alpha)\boldsymbol{\Sigma}^z$) can then be understood as ridge shrinkage as in equation 4.

### 4.2 Challenges of comparing $\mathrm{H}_\alpha(\boldsymbol{f})$ across source models/layers

Next, we discuss challenges of using $\mathrm{H}_\alpha(f)$ on the feature embeddings of target data derived from the source model for source model selection. The feature dimension ($d$) across different source models even for the penultimate layer can vary significantly e.g. from 1024 in MobileNet to 4096 in VGG19. Such differences makes source model/layer selection for fine-tuning highly problematic.

We propose dimensionality reduction of feature embeddings before computing $\mathrm{H}_\alpha(f)$. We project feature embeddings to a lower $q$-dimensional space, where $q$ is taken to be the same across the $K$ candidate models/layers and satisfies: $q \leq \min_{\boldsymbol{f}^{(1)},\boldsymbol{f}^{(2)},\cdots,\boldsymbol{f}^{(K)}} |\boldsymbol{f}^{(.)}|$ where $|.|$ operator denotes the cardinality of the feature spaces. The dimensionality reduction allows for more meaningful comparison of $\mathrm{H}_\alpha(f)$ across source/target pairs; this is relevant for source/layer selection. More generally, it also allows for faster and more robust estimate for limited target samples case ($n_t < d$) for linear and nonlinear fine-tuning. In the case of nonlinear fine-tuning, the intermediate layers of visual and language models have really large $d \sim 10^5$, see Table S1 in Supplement for examples.

We consider Gaussian Random Projection, which uses a linear transformation matrix $\boldsymbol{V}$ to derive the transformed features $\hat{\boldsymbol{F}} = \boldsymbol{F}\boldsymbol{V}$; it samples components from $\mathcal{N}(0, \frac{1}{q})$ to preserve pairwise distances between any two samples of the dataset. Untrained auto-encoders (AE) are other alternatives that have been used to detect covariate shifts in input distributions by [24]. It is not known how sensitive these untrained AE are to the underlying architecture—using trained AE is less appropriate for use in transferability measurement for fine-tuning as those maybe more time-consuming than the actual fine-tuning. We demonstrate improved correlation performance of $\mathrm{H}_\alpha(f)$ with dimensionality reduction in Table 3 in Section 6.1 for source model selection.

### 4.3 Efficient Computation for small target data

For small target data ($C \leq n_t < d$), the naive implementation of $\mathrm{H}_\alpha(f)$ can be very slow. We propose an optimized implementation for our shrinkage-based H-Score that exploits diagonal plus low-rank structure of $\boldsymbol{\Sigma}_\alpha^{(f)}$ for efficient matrix inversion and the low-rank structure of $\boldsymbol{\Sigma}^{(z)}$ for faster matrix-matrix multiplications. We assume $\boldsymbol{F}$ (and correspondingly $\boldsymbol{Z}$) are centered. The optimized computation of $\mathrm{H}_\alpha(f)$ is given by:

$$\mathrm{H}_\alpha(f) = (1-\alpha)/(n_t\alpha\sigma) \cdot \left( \|\boldsymbol{R}\|_F^2 - (1-\alpha) \cdot \mathrm{vec}\left(\boldsymbol{G}\right)^T \mathrm{vec}\left(\boldsymbol{W}^{-1}\boldsymbol{G}\right) \right), \quad (11)$$

where $\boldsymbol{R} = \left[\sqrt{n_1}\bar{\boldsymbol{f}}_{Y=1}, \cdots, \sqrt{n_C}\bar{\boldsymbol{f}}_{Y=C}\right] \in \mathbb{R}^{d\times C}$, $\boldsymbol{G} = \boldsymbol{F}\boldsymbol{R} \in \mathbb{R}^{n_t\times C}$, $\boldsymbol{W} = n_t\alpha\sigma\boldsymbol{I}_n + (1-\alpha)\boldsymbol{F}\boldsymbol{F}^T \in \mathbb{R}^{n_t\times n_t}$. The algorithm (and derivation) is provided in the Supplementary document. We make a timing comparison of our optimized implementation of $\mathrm{H}_\alpha(f)$ against the computational times of the state-of-the-art LogME measure in Section 6.3.

## 5 A closer look at NCE, LEEP and $\mathcal{N}$LEEP measures

Next, we pursue a deeper investigation of some of the newer metrics that are reported to be superior to H-Score and bring to light what appears to be some overlooked issues with these metrics in target task selection scenario. Target task selection has received less attention than source model selection. To our knowledge, we are the first to bring to light some problematic aspects with NCE, LEEP and $\mathcal{N}$LEEP, which can potentially lead to the misuse of these metrics in measuring transferability.

These measures are sensitive to the number of target classes ($C$) and tend to be smaller when $C$ is larger (see Fig. 3[Left]). Therefore, use of these measures for target tasks with *different* $C$ will most likely result in selecting the task with a smaller $C$. However, in practice, it is not always the case that transferring to a task with a smaller $C$ is easier; for example, reframing a multiclass classification

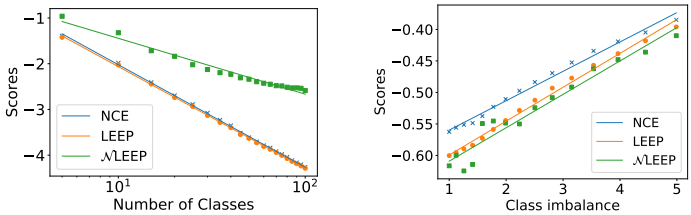


Fig. 3: Relation of NCE, LEEP & $\mathcal{N}$LEEP to [Left] number of classes (log-scale) and [Right] class imbalance, $max(n_1, n_2)/min(n_1, n_2)$, for VGG19 on CIFAR100. For [Left], we randomly select 2-100 classes. For [Right], we randomly select 2 classes and vary the class imbalances.

into a set of binary tasks can create more difficult to learn boundaries [8]. Furthermore, the measures are also problematic if two candidate target tasks have different degree of imbalance in their classes even if $C$ is the same. The measures would predict higher transferability for imbalanced data regimes over balanced settings (see Fig. 3[Right]). However, imbalanced datasets are typically harder to learn. If these measures are correlated against vanilla accuracy, which tends to be higher as the imbalance increases e.g. for binary classification, the measures would falsely suggest they are good indicators of performance. Earlier work did not consider both these aspects and erroneously showed good correlation of these metrics against vanilla accuracy to show dominance of these metrics in target task selection with different $C$ [21,30] and imbalance [30].

Here, we propose a method to ameliorate the shortcomings of NCE, LEEP and $\mathcal{N}$LEEP to prevent misuse of these measures, so that they lead to more reliable conclusions. We propose to standardize the metrics by the entropy of the target label priors, leading to the definitions in equation 12. This standardization considers both the class imbalance as well as number of classes through the entropy of the target label priors.

$$\begin{aligned} \text{n-NCE} &\stackrel{\text{def}}{=} 1 + \text{NCE}/\text{H}(Y), \\ \text{n-LEEP} &\stackrel{\text{def}}{=} 1 + \text{LEEP}/\text{H}(Y), \\ \text{n-}\mathcal{N}\text{LEEP} &\stackrel{\text{def}}{=} 1 + \mathcal{N}\text{LEEP}/\text{H}(Y). \end{aligned} \tag{12}$$

Our proposed normalizations in equation 12 ensures the normalized NCE is bounded between $[0, 1]$. For proof, see Supplementary document. n-NCE is in fact equivalent to normalized mutual information and has been extensively used to measure correlation between two different labelings/clustering of samples [32]. Given the similar behavior of LEEP and NCE to different $C$ and class imbalance as shown in Fig. 3, we suggest the same normalization as given in equation 12. However, this normalization does not ensure boundedness of n-LEEP score (and by extension n-$\mathcal{N}$LEEP) in the range $[0, 1]$ as in the case of n-NCE.

For scenarios where candidate target tasks have different $C$, we propose an alternative evaluation criteria (*relative* accuracy) instead of vanilla accuracy — see Section 6 for more details. We provide empirical validation of the proposed normalization to these measures in Table 2 in Section 6.1. We also show that our proposed shrinkage-based H-Score is the leading metric even in these scenarios.

## 6 Experiments

We evaluate existing transferability measures and our proposed modifications on two class of models: vision models and graph neural networks. We study various fine-tuning regimes and data settings. We draw inspiration from [21] who consider target task selection and source model selection. The experimental setup highlights important aspects of transferability measures, e.g., dataset size for computing empirical distributions and covariance matrices, number of target

classes, and feature dimension etc. Some of these aspects have been overlooked when evaluating transferability measures, leading to improper conclusions.

***Evaluation criteria*** Transferability measures are often evaluated by how well they correlate with the test accuracy after fine-tuning the model on target data. Following [31,21,12], we used Pearson correlation. We include additional results with respect to rank correlations (e.g., Spearman) in Supplementary document. We argue that considering correlation with the target test accuracy is flawed in some scenarios. In particular, for target task selection, it is wrong to compare target tasks based on accuracy when $C$ is different e.g 5 vs 10 classes. In such a case, task with 10 classes will have a high chance of arriving at lesser test accuracy compared to that for task with 5 classes. In this case, it is more appropriate to consider the gain in accuracy achieved by the model over it's random baseline. Hence we use relative accuracy (for balanced classes): $\frac{\text{Accuracy}-1/C}{1/C}$. This measure is more effective in capturing the performance gain achieved by the same model in transferring to two domains with different $C$. This also highlights the limitation of NCE, LEEP and $\mathcal{N}$LEEP which are sensitive to $C$ and tend to have smaller values with higher $C$; these measures do not provide useful information about how hard these different tasks when evaluated with vanilla accuracy.

Correlations marked with asterisks (*) in Tables 1, 2, 3, 4 are not statistically significant ($p$-value $> 0.05$). Larger correlations indicate better identifiability as quantified by transferability measure. Hyphen (-) indicates the computation ran out of memory or was really slow.

### 6.1 Case Study: Vision Models

First, we evaluate our proposals on visual classification tasks with vision models e.g., VGG19 [29], ResNet50 [11] that have been pre-trained on ImageNet. We fine-tune on subsets of CIFAR-100/CIFAR-10 data. We use Tensorflow Keras [3] for our implementation. Imagenet checkpoints come from Keras[4].

***Fine-tuning with hyperparameter optimization*** The optimal choice of hyperparameters for fine-tuning is not only target data dependent but also sensitive to the domain similarity between the source and target datasets [17]. We thus tune the hyperparameters for fine-tuning: we use Adam optimizer and tune batch size, learning rate, number of epochs and weight decay (L2 regularization on the classifier head). For validation tuning, we set aside a portion of the training data (20%) and try 100 random draws from hyperparameters' multi-dimensional grid. With this additional tuning complexity, we performed $650 \times 100$ fine-tuning experiments. See additional information and motivation in Supplement.

**Target Task Selection** We first investigate the correlation performance of our proposed estimator and existing transferability measures in the context of target task selection. We consider two small target data cases. We provide a summary

[4] `https://keras.io/api/applications/`

Table 1: Pearson correlation of transferability measures against fine-tuned target accuracy of vision models in the context of target task selection. We compare our proposed $H_\alpha(f)$ against original $H(f)$ and state-of-the-art measures.

| Strategy | Target | Model | Reg. | $H(f)$ | $H_\alpha(f)$ | NCE | LEEP | $\mathcal{N}$LEEP | TransRate | LFC | LogME |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LFT | CIFAR-100 | VGG19 | S-B | -0.14* | 0.81 | 0.67 | 0.65 | 0.81 | 0.56 | 0.76 | **0.85** |
| | | | S-IB | 0.03* | **0.77** | 0.57 | 0.63 | 0.70 | 0.46 | 0.47 | 0.75 |
| | | ResNet50 | S-B | 0.03* | **0.87** | 0.66 | 0.68 | 0.81 | 0.27 | 0.77 | 0.83 |
| | | | S-IB | -0.10 | 0.79 | 0.56 | 0.57 | 0.70 | 0.44 | 0.52 | **0.82** |
| | CIFAR-10 | VGG19 | S-B | 0.00* | 0.67 | 0.52 | 0.60 | 0.61 | 0.42 | 0.44 | **0.74** |
| | | | S-IB | 0.09* | 0.81 | 0.75 | 0.82 | 0.83 | 0.29 | 0.32 | **0.89** |
| | | ResNet50 | S-B | -0.29 | **0.73** | 0.43 | 0.44 | 0.61 | -0.02* | 0.57 | 0.71 |
| | | | S-IB | 0.17* | **0.89** | 0.66 | 0.71 | 0.75 | 0.28 | 0.01* | 0.83 |
| NLFT | CIFAR-100 | VGG19 | S-B | 0.17* | **0.73** | 0.58 | 0.59 | 0.67 | -0.03* | 0.70 | - |
| | | | S-IB | 0.03* | 0.49 | 0.49 | 0.54 | **0.55** | 0.48 | 0.17 | - |

of the different cases (inspired from [21]) below. Additional details are in the Supplementary document.

- *Small-Balanced Target Data (S-B):* We make a random selection of 5 classes from CIFAR-100/CIFAR-10 and sample 50 samples per class from the original train split. We repeat this exercise 50 times (with a different selection of 5 classes). We evaluate correlations of transferability measures across the 50 random experiments.
- *Small-Imbalanced Target Data (S-IB):* We make 50 random selections of 2 classes from CIFAR-100/CIFAR-10, sample between $30-60$ samples from the first class and sample $5\times$ the number of samples from the second class. This makes for a binary imbalanced classification task. We again measure performance of transferability measures against optimal target test accuracy. Note the imbalance is constant across the candidate target tasks.

We evaluate target task selection for linear and nonlinear fine-tuning under small sample setting. The layers designated as embedding layers for nonlinear fine-tuning of VGG19 and ResNet50 is given in Table S1 in Supplementary document. We empirically compare the shrinkage-based H-score against the original measure by [1] and the state-of-the-art measures. Table 1 demonstrates 80% absolute gains in correlation performance of $H_\alpha(f)$ over $H(f)$, making it a leading metric in many cases in small target data regimes.

Table 2: Pearson correlation of transferability measures against *relative* accuracy for large balanced CIFAR-100 dataset with different number of classes across target tasks.

| Model | $H(f)$ | $H_\alpha(f)$ | NCE | n-NCE | LEEP | n-LEEP | $\mathcal{N}$LEEP | n-$\mathcal{N}$LEEP | TransRate | LogME |
|---|---|---|---|---|---|---|---|---|---|---|
| VGG19 | 0.88 | **0.97** | -0.95 | 0.66 | -0.95 | 0.66 | -0.93 | 0.95 | 0.68 | 0.96 |
| ResNet50 | 0.95 | **0.98** | -0.95 | -0.74 | -0.95 | -0.73 | -0.94 | -0.63 | 0.56 | 0.96 |

Table 3: Pearson correlation of proposed $H_\alpha(f)$ without/with Random Projection (RP) for fine-tuning in source model selection of vision models in small data regimes.

| Strategy | Target | $H(f)$ | $H_\alpha(f)$[No RP] | $H_\alpha(f)$[RP] | NCE | LEEP | $\mathcal{N}$LEEP | TransRate | LogME |
|---|---|---|---|---|---|---|---|---|---|
| LFT | CIFAR-100 | -0.190* | 0.024* | **0.859** | 0.825 | 0.839 | 0.852 | -0.204* | 0.705 |
| | CIFAR-10 | 0.276* | 0.277* | **0.939** | 0.938 | 0.936 | 0.938 | 0.311* | 0.923 |
| NLFT | CIFAR-100 | -0.108* | 0.125* | 0.879 | 0.967 | 0.976 | **0.977** | - | - |

Next, we study a large-balanced target data setting where the number of classes varies across the target tasks. We construct the target tasks as follows: We randomly select 2-100 classes from CIFAR-100 and include all samples from the chosen classes. This constructs a collection of large balanced target tasks with different number of classes (L-B-C). We generate 50 such target datasets. In this setting, we evaluate correlation of transferability measures with *relative* target test accuracy for reasons highlighted in Section 5. This setting validates that the normalizations, proposed in equation 12 in Section 5, can improve the correlations of NCE, LEEP and $\mathcal{N}$LEEP when evaluated on the more appropriate relative accuracy scale. Table 2 demonstrates how various transferability measures perform on target task selection when the number of target classes **varies**. $H_\alpha(f)$ dominates the performance for both VGG19 and ResNet50 models, surpassing all transferability measures, closely followed by LogME.

**Source Model Selection** Next, we study source model selection scenario for vision models. We select 9 small to large (pre-trained ImageNet) vision models: VGG19, ResNet50, ResNet101, DenseNet121, DenseNet201, Xception, InceptionV3, MobileNet, EfficientNetB0. We evaluate source model selection for linear and nonlinear fine-tuning under small sample setting. The layers designated as embedding layers for nonlinear fine-tuning of all 9 models is shown in Table S1 in Supplementary document. We sample 50 images per class from all classes available in the original train split of CIFAR-100/CIFAR-10. We designate 10 samples per class for hyperparameter tuning.

We demonstrate that $H_\alpha(f)$ is a leading metric in source model selection as well. Given that the feature dimensions vary significantly across different models in source model selection for both linear and nonlinear finetuning, we apply proposed dimensionality reduction via random projection in Section 4.2 for $H_\alpha(f)$ to project feature embeddings to 128-dimensional space ($q = 128$). This allows for more meaningful comparison of H-score across source models. This leads to the gains of proposed $H_\alpha(f)$ in terms of correlation in the context of source model selection as well for small samples as given in Table 3, making it again a leading metric in source model selection.

### 6.2 Case Study: Graph Neural Networks

We evaluate our proposals on Graph Neural Networks on Twitch Social Networks [26,25,27]. The datasets are social networks of gamers from the streaming service Twitch, where nodes correspond to Twitch users and links correspond to mutual friendships. There are country-specific sub-networks. We consider six sub-networks: {DE, ES, FR, RU, PTBR, ENGB}. Features describe the history of games played and the associated task is binary classification of whether a gamer streams adult content. The country specific graphs share the same node features which means that we can perform transfer learning with these datasets. Additional details about the Twitch Social Networks datasets are included in Supplementary document.

***Transferability Setup*** We consider a two-layered Graph Convolutional Network (GCN) [13]. The network takes the following functional form:

$$\text{logit}(\boldsymbol{x}) = \hat{\boldsymbol{G}} \cdot \text{Dropout}(\text{ReLU}(\hat{\boldsymbol{G}} \cdot \boldsymbol{x} \cdot \boldsymbol{W}^1)) \cdot \boldsymbol{W}^2, \tag{13}$$

where $\hat{\boldsymbol{G}} = \hat{\boldsymbol{D}}^{1/2}(\boldsymbol{G} + \boldsymbol{I})\hat{\boldsymbol{D}}^{1/2}$ denotes the renormalization trick from [13] when applied to the graph adjacency matrix $\boldsymbol{G} \in \mathbb{R}^{m \times m}$, $\hat{D}_{ii} = \sum_j (\boldsymbol{G} + \boldsymbol{I}_m)_{ij}$ denotes the degree of node $i$, $\boldsymbol{W}^1 \in \mathbb{R}^{p \times d}$ and $\boldsymbol{W}^2 \in \mathbb{R}^{d \times C}$ denote the learnable weights for the first and second layer respectively. The mapping from logits to $Y$ can be done by applying a softmax and returning the class with the highest probability.

For studying transferability, we consider the target feature embeddings as: $\boldsymbol{F} = h(\boldsymbol{X}^{(t)}) = \hat{\boldsymbol{G}} \cdot \text{Dropout}(\text{ReLU}(\hat{\boldsymbol{G}} \cdot \boldsymbol{X}^{(t)} \cdot \boldsymbol{W}^1))$. This creates a linear transfer learning strategy, which is similar to the linear fine-tuning regime studied for vision models in section 6.1. We study target task selection in Section 6.2 and source model selection in Supplementary document.

***Pre-training Implementation*** We use PyTorch Geometric [7] to setup the pre-training of GCN models. The pre-training uses a country-specific subnetwork and performs training, model selection and testing via a 64%/16%/20% split. The training considers transductive learning in graph networks. We perform 200 hyperparameter trials that tune over Adam learning rates $[10^{-5}, 10^{-1}]$, batch sizes $\{16, 32, 64\}$, L2 regularization $[10^{-4}, 1]$, dropout of 0.5 and maximum 1000 epochs with early stopping with a patience of 50.

***Fine-tuning*** For fine-tuning experiments, we recover the target feature embeddings by applying optimal pre-trained source model on a different subnetwork of users. Given we study linear fine-tuning in Graph Networks, we use GridSearchCV with $\ell_2$-regularized Logistic Regression from sklearn [22] on (only) target train data to perform (stratified) 5-fold cross-validation. We consider 100 values for L2 regularization in the range $[10^{-5}, 10^3]$ on the log scale. The optimal L2 regularization is used to get the optimal model and the test accuracy is computed to measure correlation of transferability measures.

**Target Task Selection** We pre-train the GCN model with each of the country-specific sub-network. For each country-specific sub-network, $S \in \{$DE, ES, FR,

Table 4: Pearson correlation of transferability measures against fine-tuned target accuracy of *Graph Convolutional Networks* in Target Task selection scenario. We compare our proposed $\mathrm{H}_\alpha(f)$ against original $\mathrm{H}(f)$ and state-of-the-art measures.

| $n_t$ | Model | Source | $\mathrm{H}(f)$ | $\mathrm{H}_\alpha(f)$[5] | NCE | LEEP | $\mathcal{N}$LEEP | LFC | TransRate | LogME |
|---|---|---|---|---|---|---|---|---|---|---|
| 500 | GCN-256 | DE | 0.10* | 0.35 | 0.15* | 0.15* | 0.40 | **0.53** | -0.34 | 0.40 |
| | | ES | 0.24 | 0.41 | 0.48 | **0.50** | -0.13* | 0.24* | -0.34* | 0.20* |
| | | FR | 0.57 | **0.61** | -0.03* | 0.01* | -0.05* | 0.26* | -0.30* | 0.44 |
| | | RU | 0.11* | **0.34** | -0.19* | -0.17* | -0.12* | 0.04* | -0.21 | 0.11* |
| | | PTBR | **0.37** | 0.24* | 0.16* | 0.16* | -0.13* | -0.02* | -0.05* | 0.15* |
| | | ENGB | 0.48 | **0.53** | -0.12* | -0.09* | -0.12* | 0.15* | -0.05* | 0.32* |
| 1000 | GCN-512 | DE | 0.48 | **0.71** | 0.29* | 0.44 | -0.14* | 0.45 | 0.17* | 0.59 |
| | | ES | 0.68 | **0.78** | 0.59 | 0.54 | 0.35* | 0.49 | -0.12* | 0.59 |
| | | FR | 0.59 | **0.61** | 0.25* | 0.27* | 0.04* | 0.10* | 0.58 | 0.22* |
| | | RU | 0.35 | **0.44** | -0.12* | 0.08* | -0.25* | 0.14* | -0.07* | 0.27* |
| | | PTBR | 0.67 | **0.77** | 0.37 | 0.40 | 0.04* | 0.18* | 0.10 | 0.50 |
| | | ENGB | **0.82** | 0.81 | -0.04* | -0.08* | 0.17* | 0.26* | 0.15* | 0.65 |

RU, PTBR, ENGB}, we construct 30 different target tasks. We exclude the specific country on which the source model is pre-trained and use the remaining countries to construct different combinations of networks as targets. For example, if the source model is pre-trained on DE, then the target tasks are given by: $\{\text{ES}, \text{FR}, \text{RU}, \text{PTBR}, \text{ENGB}, (\text{ES,FR}), (\text{ES,RU}), \cdots, (\text{ES,FR,RU,PTBR,ENGB})\}$.

We study balanced targets in this regime. Given that the degree of imbalance varies significantly across different country-specific networks, we collect the largest balanced datasets for each target. Next, we sample $n_t = 1000$ nodes for fine-tuning and allocate the remaining nodes as test samples. We consider two different embedding sizes for GCN network in this study. We also validate our proposals when we allocate 500 samples for fine-tuning.

We present the Pearson correlation performance of transferability measures against fine-tuned target test accuracy in Table 4. The correlations demonstrate our proposed $\mathrm{H}_\alpha(f)$ as the leading metric for target task selection for linear fine-tuning in Graph Neural Networks. We include additional results for source model selection in Supplementary document.

### 6.3 Timing comparison between LogME and $\mathrm{H}_\alpha(f)$

We empirically investigate the computational times of $\mathrm{H}_\alpha(f)$ when computed via our optimized implementation in equation 11. For this exercise, we generate synthetic multi-class classification data using Sklearn [22] multi-class dataset generation function that is adapted from [9]. We investigate different values for number of samples ($n_t$), feature dimension ($d$) and number of classes ($C$). For

[5] We empirically observed dimensionality reduction with random projection to improve correlation performance for $\mathrm{H}_\alpha(f)$ in this target task selection setting as well. We used $q = 128$.

Table 5: Timing comparison of LogME and our $\mathrm{H}_\alpha(f)$. All times are in $ms$.

| $n_t$ | $d$ | $\lvert\mathcal{Y}\rvert = C$ | LogME | $\mathrm{H}(f)$ | $\mathrm{H}_\alpha(f)$ |
|---|---|---|---|---|---|
| 500 | 500 | 50 | 201 | 123 | **22** |
| 500 | 1000 | 50 | 185 | 376 | **36** |
| 500 | 5000 | 50 | 392 | 9680 | **373** |
| 500 | 1000 | 10 | 111 | 259 | **33** |
| 500 | 1000 | 100 | 268 | 271 | **36** |
| 100 | 1000 | 50 | 72 | 255 | **16** |
| 1000 | 1000 | 50 | 318 | 335 | **75** |

data generation, we set number of informative features to be 100 with the rest of the features filled with random noise. For LogME, we use a faster variant proposed by [34]. For a fair comparison, we do not use any dimensionality reduction in the computation of $\mathrm{H}_\alpha(f)$. Table 5 demonstrates a significant computational advantage of $\mathrm{H}_\alpha(f)$ over LogME. We observe $3-10$ times faster computational times.

## 7 Conclusion

We study transferability measures in the context of fine-tuning. Our contributions are three-fold. First, we show that H-score measure, commonly used as a baseline for newer transferability measures, suffers from instability due to poor estimation of covariance matrices. We propose shrinkage-based estimation of H-score with regularized covariance estimation techniques from statistical literature. We show 80% absolute increase over the original H-score and show superior performance in many cases against all newer transferability measures across various model types, fine-tuning scenarios and data settings. Second, we present a fast implementation of our estimator that provides a $3-10$ times computational advantage over state-of-the-art LogME measure. Third, we identify problems with 3 other transferability measures (NCE, LEEP and $\mathcal{N}$LEEP) in target task selection (an understudied fine-tuning scenario than source model selection) when either the number of target classes or the class imbalance varies across candidate target tasks. We propose an alternative evaluation scheme that measures correlation against relative target accuracy (instead of vanilla accuracy) in such scenarios. Our large set of $\sim 164,000$ fine-tuning experiments with multiple vision models and graph neural networks in different regimes demonstrates usefulness of our proposals. We leave it for future work to explore how predictive various transferability measures are for co-training regimes (as opposed to fine-tuning).

## References

1. Bao, Y., Li, Y., Huang, S., et al.: An information-theoretic approach to transferability in task transfer learning. In: 2019 IEEE ICIP. pp. 2309–2313 (2019)

2. Chen, Y., Wiesel, A., Eldar, Y.C., et al.: Shrinkage algorithms for mmse covariance estimation. IEEE Transactions on Signal Processing **58**(10), 5016–5029 (2010)
3. Chollet, F., et al.: Keras. `https://github.com/fchollet/keras` (2015)
4. Cui, Y., Song, Y., Sun, C., et al.: Large scale fine-grained categorization and domain-specific transfer learning. CoRR **abs/1806.06193** (2018)
5. Deshpande, A., Achille, A., Ravichandran, A., et al.: A linearized framework and a new benchmark for model selection for fine-tuning (2021)
6. Devlin, J., Chang, M., Lee, K., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
7. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
8. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). The Annals of Statistics **28**(2), 337 – 407 (2000)
9. Guyon, I.: Design of experiments for the nips 2003 variable selection benchmark (2003)
10. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA (2001)
11. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015)
12. Huang, L.K., Wei, Y., Rong, Y., et al.: Frustratingly easy transferability estimation. ArXiv **abs/2106.09362** (2021)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR 2017, Toulon, France, April 24-26. OpenReview.net (2017)
14. Kornblith, S., Shlens, J., Le, Q.V.: Do better imagenet models transfer better? In: 2019 IEEE/CVF CVPR. pp. 2656–2666 (2019)
15. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems **25** (01 2012)
16. Ledoit, O., Wolf, M.: A well-conditioned estimator for large-dimensional covariance matrices. Journal of Multivariate Analysis **88**(2), 365–411 (Feb 2004)
17. Li, H., Chaudhari, P., Yang, H., et al.: Rethinking the hyperparameters for fine-tuning. CoRR **abs/2002.11770** (2020)
18. Li, Y., Jia, X., Sang, R., et al.: Ranking neural checkpoints. In: Proceedings of the IEEE/CVF CVPR. pp. 2663–2673 (June 2021)
19. Mahajan, D., Girshick, R.B., Ramanathan, V., et al.: Exploring the limits of weakly supervised pretraining. CoRR **abs/1805.00932** (2018)
20. Max, A.W.: Inverting modified matrices. In: Memorandum Rept. 42, Statistical Research Group, p. 4. Princeton Univ. (1950)
21. Nguyen, C.V., Hassner, T., Seeger, M., Archambeau, C.: Leep: A new measure to evaluate transferability of learned representations (2020)
22. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12**(null), 2825-2830 (Nov 2011)
23. Pourahmadi, M.: High-dimensional covariance estimation: with high-dimensional data, vol. 882. John Wiley & Sons (2013)
24. Rabanser, S., Günnemann, S., Lipton, Z.C.: Failing loudly: An empirical study of methods for detecting dataset shift. In: NeurIPS (2019)
25. Rozemberczki, B., Allen, C., Sarkar, R.: Multi-Scale Attributed Node Embedding. Journal of Complex Networks **9**(2) (2021)
26. Rozemberczki, B., Sarkar, R.: Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In: Proceedings of the 29th ACM CIKM. p. 1325-1334. CIKM '20, ACM, New York, NY, USA (2020)

27. Rozemberczki, B., Sarkar, R.: Twitch gamers: a dataset for evaluating proximity preserving and structural role-based node embeddings (2021)
28. Schäfer, J., Strimmer, K.: A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical Applications in Genetics and Molecular Biology **4** (2005)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2015)
30. Tan, Y., Li, Y., Huang, S.: OTCE: A transferability metric for cross-domain cross-task representations. CoRR **abs/2103.13843** (2021)
31. Tran, A., Nguyen, C., Hassner, T.: Transferability and hardness of supervised classification tasks. In: 2019 IEEE/CVF ICCV. pp. 1395–1405 (2019)
32. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research **11**(95), 2837–2854 (2010)
33. You, K., Liu, Y., Wang, J., Long, M.: Logme: Practical assessment of pre-trained models for transfer learning. In: ICML (2021)
34. You, K., Liu, Y., Zhang, Z., Wang, J., Jordan, M.I., Long, M.: Ranking and tuning pre-trained models: A new paradigm of exploiting model hubs (2021)