# LEEP: A New Measure to Evaluate Transferability of Learned Representations

Cuong V. Nguyen [1]   Tal Hassner [2]   Matthias Seeger [1]   Cedric Archambeau [1]

## Abstract

We introduce a new measure to evaluate the transferability of representations learned by classifiers. Our measure, the *Log Expected Empirical Prediction* (LEEP), is simple and easy to compute: when given a classifier trained on a source data set, it only requires running the target data set through this classifier once. We analyze the properties of LEEP theoretically and demonstrate its effectiveness empirically. Our analysis shows that LEEP can predict the performance and convergence speed of both transfer and meta-transfer learning methods, even for small or imbalanced data. Moreover, LEEP outperforms recently proposed transferability measures such as negative conditional entropy and H scores. Notably, when transferring from ImageNet to CIFAR100, LEEP can achieve up to 30% improvement compared to the best competing method in terms of the correlations with actual transfer accuracy.

## 1. Introduction

Transferability estimation (Eaton et al., 2008; Ammar et al., 2014; Sinapov et al., 2015) is the problem of quantitatively estimating how easy it is to transfer knowledge learned from one classification task to another. Specifically, given a source task, represented by a labeled data set or a pre-trained model, and a target task, represented by a labeled data set, transferability estimation aims to develop a measure (or a score) that can tell us, ideally without training on the target task, how effectively transfer learning algorithms can transfer knowledge from the source task to the target task.

Answering this question is important, since good estimations of transferability can help understand the relationships between tasks (Tran et al., 2019), select groups of highly transferable tasks for joint training (Zamir et al., 2018), or

choose good source models for a given target task (Achille et al., 2019; Bao et al., 2019; Bhattacharjee et al., 2019). Previous approaches to transferability estimation often require running a transfer learning algorithm that involves expensive parameter optimization (Zamir et al., 2018; Achille et al., 2019), do not have a simple interpretation (Bao et al., 2019), or make strong assumptions about the data sets that limit their applicability (Zamir et al., 2018; Tran et al., 2019).

We propose a novel measure called the *Log Expected Empirical Prediction* (LEEP) for transferability estimation of deep networks that overcomes all the shortcomings above. In contrast to previous approaches, LEEP scores are obtained without training on the target task, thus avoiding the expensive parameter optimization step. Additionally, they have a simple interpretation and can be applied in general settings to a wide range of modern deep networks.

In particular, LEEP scores are obtained from a source model and a target data set by making a *single* forward pass of the model through the target data. This is a simpler process than previous methods, such as Taskonomy (Zamir et al., 2018) and Task2Vec (Achille et al., 2019), where one has to re-train at least part of the source model on the target data set. Furthermore, LEEP has a simple interpretation: it is the average log-likelihood of the *expected empirical predictor*, a simple classifier that makes prediction based on the expected empirical conditional distribution between source and target labels. Finally, LEEP does not make any assumption on the source and target input samples, except that they have the same size. This is more general and applicable than previous work (Zamir et al., 2018; Tran et al., 2019) where source and target data sets were assumed to share the same input samples.

**Contributions**. We formally define LEEP and rigorously analyze it, both theoretically and empirically. We show two theoretical properties of the measure: (1) LEEP is upper bounded by the average log-likelihood of the optimal model, obtained by re-training the head classifier while freezing the feature extractor; (2) LEEP is related to the negative conditional entropy measure proposed by Tran et al. (2019).

We conduct extensive experiments to evaluate our LEEP measure in several scenarios. We show that the measure is useful for predicting the performance of two com-

monly used transfer learning algorithms – head classifier re-training (Donahue et al., 2014; Razavian et al., 2014) and model fine-tuning (Agrawal et al., 2014; Girshick et al., 2014) – not only for large target data sets, but also for small or imbalanced target data sets that are difficult to use for re-training. We also show that LEEP can predict the convergence speed of the fine-tuning method for transfer learning.

We further demonstrate that LEEP can predict the performance of a recently developed meta-transfer learning method, the Conditional Neural Adaptive Processes (Requeima et al., 2019). Meta-transfer learning (Wei et al., 2018b; Sun et al., 2019; Requeima et al., 2019) is a framework for learning to transfer using several meta-training tasks. Importantly, to our knowledge, our work is the first to develop a transferability measure for meta-transfer learning.

We empirically compare our method with the very recent negative conditional entropy measure (Tran et al., 2019) and H scores (Bao et al., 2019). Our comparisons show that LEEP better correlates with the actual transfer accuracy than these methods. Finally, we demonstrate the effectiveness of LEEP for the source model selection problem in comparison with the negative conditional entropy and H scores.

## 2. Log Expected Empirical Prediction

Consider transfer learning between two classification tasks: a source task, represented by a pre-trained model, and a target task, represented by a labeled data set. Formally, assume the source task requires learning a model that maps input instances from a domain $\mathcal{X} = \mathbb{R}^N$ to labels in a finite label set $\mathcal{Z}$. Further, assume that we already trained such a model, which we call the *source model*, denoted by $\theta$. We note that $\mathcal{X}$ can contain text or images since they can be flatten to vectors in $\mathbb{R}^N$. Transfer learning seeks to learn a model for a target task mapping inputs from $\mathcal{X}$ to some finite target label set $\mathcal{Y}$, given a *target data set* $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, for this purpose. We emphasize that, unlike recent previous work (Zamir et al., 2018; Tran et al., 2019), the domain $\mathcal{X}$ in our setting is general: the input instances of the source and target tasks are only assumed to have the same dimension,[1] such as ImageNet (Russakovsky et al., 2015) and CIFAR (Krizhevsky, 2009) images scaled to the same size.

The transfer learning setting above is very common in computer vision for leveraging expensive pre-trained deep learning models. For instance, a popular transfer learning method takes a model, pre-trained on ImageNet, and re-trains its classifier (the head) on the target data set while freezing the feature extractor. The result is a new model for the target

task that uses the representation learned on the source with a head classifier learned on the target (Donahue et al., 2014; Razavian et al., 2014; Zeiler & Fergus, 2014; Oquab et al., 2014; Whatmough et al., 2019). We call this the *head retraining* method. Another popular transfer learning method, called the *fine-tuning* method, fine-tunes the feature extractor while re-training the new head classifier on the target data set to get a model for the target task (Agrawal et al., 2014; Girshick et al., 2014; Chatfield et al., 2014; Dhillon et al., 2020).

In this work, we study the *transferability estimation* problem, which aims to develop a measure (or a score) that can tell us, without training on the target data set, how effectively these transfer learning algorithms can transfer knowledge learned in the source model $\theta$ to the target task, using the target data set $\mathcal{D}$. We emphasize that although transferability estimation can tell us the effectiveness of transfer learning, it is not a solution for transfer learning in itself. Instead, transfer learning is achieved by e.g., the head retraining or fine-tuning methods above. In our experiments in Sec. 5, we will use these transfer learning methods to test our transferability measure.

We now describe our proposed measure, LEEP, that requires no expensive training on the target task and offers an *a priori* estimate of how well a model will transfer to the target task. The measure can be efficiently computed from the source model $\theta$ and the target data set $\mathcal{D}$. The computation involves the following three steps.

**Step 1: Compute dummy label distributions of the inputs in the target data set $\mathcal{D}$.** We apply $\theta$ to each input $x_i$ in $\mathcal{D}$ to get the predicted distribution over $\mathcal{Z}$, the label set of the source task. We denote this predicted distribution by $\theta(x_i)$, which is a categorical distribution over $\mathcal{Z}$. Note that $\theta(x_i)$ is a *dummy distribution* over labels of the source task, since these labels may not be meaningful for the example $x_i$. For instance, if $\theta$ is a model pre-trained on ImageNet and $\mathcal{D}$ is the CIFAR data set, then $\theta(x_i)$ is a distribution over ImageNet labels, which may not be semantically related to the true label of $x_i$ in the CIFAR data set.

**Step 2: Compute the empirical conditional distribution $\hat{P}(y|z)$ of the target label $y$ given the source label $z$.** We next compute the empirical conditional distribution $\hat{P}(y|z)$ for all $(y, z) \in \mathcal{Y} \times \mathcal{Z}$. To this end, we first compute the empirical joint distribution $\hat{P}(y, z)$ for all label pairs $(y, z) \in \mathcal{Y} \times \mathcal{Z}$:

$$\hat{P}(y, z) = \frac{1}{n} \sum_{i:y_i=y} \theta(x_i)_z, \qquad (1)$$

where the summation $\sum_{i:y_i=y}$ means we sum over all indices $i \in \{1, 2, \ldots, n\}$ that satisfy $y_i = y$. In the above equation, $\theta(x_i)_z$ is the probability of the label $z$ according to the categorical distribution $\theta(x_i)$.

---

[1]If the source model is fully convolutional (Long et al., 2015), this assumption can be relaxed.

From this empirical joint distribution, we can compute the empirical marginal distribution $\hat{P}(z)$:

$$\hat{P}(z) = \sum_{y \in \mathcal{Y}} \hat{P}(y, z) = \frac{1}{n} \sum_{i=1}^{n} \theta(x_i)_z,$$

and then the empirical conditional distribution $\hat{P}(y|z)$:

$$\hat{P}(y|z) = \frac{\hat{P}(y, z)}{\hat{P}(z)}.$$

**Step 3: Compute LEEP using $\theta(x)$ and $\hat{P}(y|z)$.** For any input $x \in \mathcal{X}$, consider a classifier that predicts a label $y$ of $x$ by first randomly drawing a dummy label $z$ from $\theta(x)$ and then randomly drawing $y$ from $\hat{P}(y|z)$. Equivalently, this classifier can predict $y$ by directly drawing a label from the distribution $p(y|x; \theta, \mathcal{D}) = \sum_{z \in \mathcal{Z}} \hat{P}(y|z) \, \theta(x)_z$. We call this classifier the *Expected Empirical Predictor* (EEP).

For the target data set $\mathcal{D}$, we define LEEP as the average log-likelihood of the EEP classifier given the data $\mathcal{D}$. Formally, our LEEP measure of transferability is defined as:

$$T(\theta, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \log \left( \sum_{z \in \mathcal{Z}} \hat{P}(y_i|z) \, \theta(x_i)_z \right). \quad (2)$$

Intuitively, this measure tells us how well the EEP performs on $\mathcal{D}$. We argue that since the EEP is constructed mainly from the source model $\theta$ with a minimal use of $\mathcal{D}$ (i.e., $\mathcal{D}$ is only used to compute the simple empirical conditional distribution), it can serve as an indicator of how "close" $\theta$ and $\mathcal{D}$ are, and hence a measure of transferability.

From its definition, the LEEP measure is always negative and larger values (i.e., smaller absolute values) indicate better transferability. When the target task contains more classes, LEEP scores tend to be smaller. The measure is also efficient to compute since its computational bottleneck (step 1 above) requires only a single forward pass through the target data set $\mathcal{D}$.

## 3. Theoretical Properties of LEEP

Assume $\theta = (w, h)$ where $w$ is a feature extractor that maps an input $x \in \mathcal{X}$ to a representation (or *embedding*), $r = w(x)$, and $h$ is a classifier (or *head*) that takes the representation $r$ as input and returns a probability distribution over $\mathcal{Z}$. Next, assume we fix $w$ and re-train the classifier by maximum likelihood, using the target data set $\mathcal{D}$, to obtain the new classifier $k^*$. That is,

$$k^* = \arg \max_{k \in \mathcal{K}} \, l(w, k), \quad (3)$$

where $l(w, k) = \frac{1}{n} \sum_{i=1}^{n} \log p(y_i|x_i; w, k)$ is the average log-likelihood of $(w, k)$ on the target data set $\mathcal{D}$, and $\mathcal{K}$ is the space of classifiers from which we would like to select $k$.

We note that $\mathcal{K}$ contains classifiers that map $w(x)$ to labels in $\mathcal{Y}$. If we assume $\mathcal{K}$ contains the EEP, we can easily show the following property of the LEEP measure (see proof in the supplementary material):

**Property 1.** *LEEP is a lower bound of the optimal average log-likelihood. Formally, $T(\theta, \mathcal{D}) \leq l(w, k^*)$.*

The assumption that $\mathcal{K}$ contains the EEP can be easily satisfied if we select $\mathcal{K} = \bar{\mathcal{K}} \cup \{k_{\text{EEP}}\}$, where the classifier $k_{\text{EEP}}$ is the EEP and $\bar{\mathcal{K}}$ is a space of classifiers that we can easily optimize over (e.g., the parameter space of a linear classifier). We can then solve Eq. (3) by the following two-stage process. First, we solve $\bar{k} = \arg \max_{k \in \bar{\mathcal{K}}} \, l(w, k)$ using, for instance, stochastic gradient descent (SGD) (Robbins & Monro, 1951; Bottou, 1991). Then, we solve $k^* = \arg \max_{k \in \{\bar{k}, k_{\text{EEP}}\}} l(w, k)$ by simple comparison. In our experiments, we skip this second step and choose $k^* = \bar{k}$, as usually done in practice. The results reported in Sec. 5.1 show that LEEP correlates well with the test accuracy of $\bar{k}$, indicating that the measure can be used to estimate the performance of the transferred model in practice.

As a second property, we show a relationship between LEEP and the negative conditional entropy (NCE) measure recently proposed by Tran et al. (2019). To measure the transferability between $\theta$ and $\mathcal{D}$, we can compute the NCE measure as follows. First, we use $\theta$ to label all $x_i$'s in $\mathcal{D}$. For example, we can label $x_i$ by a dummy label $z_i = \arg \max_{z \in \mathcal{Z}} \theta(x_i)_z$. Then, using the dummy label set $Z = (z_1, z_2, \ldots, z_n)$ and the true label set $Y = (y_1, y_2, \ldots, y_n)$, we can compute $\text{NCE}(Y|Z)$. It can be shown that the following property relating LEEP and NCE holds (see proof in the supplementary material):

**Property 2.** *LEEP is an upper bound of the NCE measure plus the average log-likelihood of the dummy labels. Formally, $T(\theta, \mathcal{D}) \geq \text{NCE}(Y|Z) + \frac{1}{n} \sum_{i=1}^{n} \log \theta(x_i)_{z_i}$.*

From Properties 1 and 2, we know that LEEP is bounded between $\text{NCE}(Y|Z) + \frac{1}{n} \sum_{i} \log \theta(x_i)_{z_i}$ and the average log-likelihood of the re-trained model. When the re-trained model does not overfit, its average log-likelihood is a reasonable indicator of the model's performance (Tran et al., 2019). When LEEP is close to this average log-likelihood, it can be considered, in some sense, a reasonable indicator of transferability. Our experiments in Sec. 5.6 show the effectiveness of LEEP over NCE.

We note that in the setting of Tran et al. (2019), the source data set is given, instead of the source model $\theta$. Hence, NCE is a more natural measure of transferability in their case. In contrast, LEEP is more natural for our setting, since we are only given the source model $\theta$. In return, by assuming a source model, LEEP is not restricted to their setting, where the two tasks are defined over the exact same input instances.

## 4. Related Work

Our work is related to several research areas in machine learning and computer vision, including transfer learning (Weiss et al., 2016; Yang et al., 2020), meta-transfer learning (Wei et al., 2018b; Sun et al., 2019; Requeima et al., 2019), task space modeling (Zamir et al., 2018; Achille et al., 2019), and domain adaptation (Sun et al., 2016; Azizzadenesheli et al., 2019). We discuss below previous work that is closely related to ours.

**Transfer learning**. Our paper addresses the problem of predicting the performance of transfer learning algorithms between two classification tasks, without actually executing these algorithms. This problem is also called transferability estimation between classification tasks (Bao et al., 2019; Tran et al., 2019). Early theoretical work in transfer and multi-task learning studied the *relatedness* between tasks and proposed several types of distances between tasks. These distances include the $\mathcal{F}$-relatedness (Ben-David & Schuller, 2003), $\mathcal{A}$-distance (Kifer et al., 2004; Ben-David et al., 2007), and discrepancy distance (Mansour et al., 2009). Although useful for theoretical analysis, these approaches are unsuited for measuring transferability in practice because they cannot be computed easily and are symmetric. Transferability measures should be non-symmetric since transferring from one task to another (e.g., from a hard task to an easy one) is different from transferring in the reverse direction (e.g., from the easy task to the hard one).

More recently, Azizzadenesheli et al. (2019) studied domain adaptation under label shift. The authors considered the case where only the marginal label distribution changes between domains. This assumption is more restrictive than the setting in our paper as we allow both the input distribution and the label distribution to change arbitrarily. Task similarity was also considered for Gaussian process based transfer learning between regression tasks (Cao et al., 2010; Wei et al., 2018a).

The most related work to ours are transferability measures recently proposed by Tran et al. (2019) and Bao et al. (2019). Tran et al. (2019) developed the negative conditional entropy measure between the source and target label sets, under the assumption that the source and target data sets share the same input examples. Our paper removes this requirement and allows the input data to come from arbitrarily different distributions. Bao et al. (2019) developed a transferability measure based on H scores, which are derived from information-theoretic principles. Although, like us, their transferability measure can be applied to general settings, their measure is hard to interpret since it involves solving a Hirschfeld-Gebelein-Rényi maximum correlation problem (Hirschfeld, 1935; Gebelein, 1941; Rényi, 1959). LEEP scores, on the other hand, have a simple interpretation

related to the EEP and are easy to implement and compute.

**Meta-transfer learning**. Meta-transfer learning is a framework for learning to transfer from a source task to a target task (Wei et al., 2018b; Sun et al., 2019; Requeima et al., 2019). Similar to our transfer learning setting, Sun et al. (2019) and Requeima et al. (2019) also adapted a pre-trained model on the source task to the target task. These meta-learning methods learn the adaptation from several meta-training tasks, which consist of additional target data sets and target test sets from different domains that are intended to mimic the transfer learning scenario, where one wants to transfer knowledge to *unseen* tasks. Because of the additional data sets, the transfer learning mechanism in meta-learning departs from that of regular transfer learning. Nevertheless, we show that LEEP scores can also predict the performance of meta-transfer learning algorithms, such as the conditional neural adaptive processes (CNAPs) recently proposed by Requeima et al. (2019). To our knowledge, we are the first to develop a transferability measure that can be applied to meta-transfer learning.

**Task space representation**. Our paper is related to task space representation (Edwards & Storkey, 2017; Zamir et al., 2018; Achille et al., 2019; Jomaa et al., 2019) in the sense that transferability may be estimated from a distance between the representations (or embeddings) of tasks. For instance, Task2Vec (Achille et al., 2019) tried to map tasks (or data sets) to vectors in a vector space. Transferability between tasks was then estimated using a non-symmetric distance between the corresponding vectors. This method requires training a large reference network (called the probe network), adapting it to the target data set, and computing the Fisher information matrix to obtain a task embedding. Our method, on the other hand, is much simpler and computationally more efficient. Additionally, we require only the source model and a small target data set, which are both usually available in practice.

Edwards & Storkey (2017) extended the variational auto-encoder (Kingma & Welling, 2014) to compute statistics of data sets that are useful in a range of applications, including clustering data sets or classifying unseen classes. These statistics, however, were not shown to be useful for transferability estimation. Another related work, Taskonomy (Zamir et al., 2018), created a taxonomy of tasks that revealed task structures useful for reducing the number of labeled training data. Taskonomy involves several steps, one of which requires computing the task affinity matrix by re-training networks on target data sets. LEEP scores can be regarded as an efficient approximation of the task affinity, as we avoid network re-training.

# 5. Experiments

We evaluate the ability of LEEP to predict the performance of transfer and meta-transfer learning algorithms, prior to applying these algorithms in practice. We further show that LEEP is useful even in the small or imbalanced data settings, where training on the target task could be hard. We compare LEEP with the state of the art NCE transferability measure of Tran et al. (2019) and H score of Bao et al. (2019). Finally, we demonstrate the use of LEEP for source model selection. Our experiments are implemented in Gluon/MXNet (Chen et al., 2015; Guo et al., 2019).

## 5.1. LEEP vs. Transfer Accuracy

We show that LEEP scores effectively predict the accuracies of models transferred from source to target tasks. We consider two different source models, each one representing a different source task: ResNet18 (He et al., 2016), which is pre-trained on ImageNet (Russakovsky et al., 2015), and ResNet20 (He et al., 2016), which is pre-trained on CIFAR10 (Krizhevsky, 2009). For each model, we construct 200 different target tasks from the CIFAR100 data set (Krizhevsky, 2009) as follows. The label set of each target task is constructed by randomly drawing a subset of the 100 classes, with the subset's size ranging from 2 to 100. The target data set then consists of all training examples of the selected classes in the CIFAR100 data set. We use test examples of these selected classes as the target test set to compute the accuracy of the transferred model.

We experiment with two commonly used transfer learning methods to train a transferred model to a target task:

**Re-train head**. This method keeps the feature extractor layers of the source model fixed and then trains a new head classifier using the target data set from scratch. We re-train the new classifier by running SGD on the cross entropy loss.

**Fine-tune**. This method replaces the head classifier of the source model with a new head and then fine-tunes the entire model – the feature extractor and the new head – using the target data set. Fine-tuning is performed again by running SGD on the cross entropy loss.

To clarify, we let the feature extractor be the portion of the source model's network up to and including the penultimate layer. The head classifier is the network' last fully connected layer. For each target task, models transferred using these two methods were evaluated on the target test set to obtain the test accuracies. We then compare these accuracies with our LEEP scores evaluated on these target tasks.

In all tests, we ran SGD for 100 epochs with learning rate 0.01 and batch size 10 since they were sufficient to obtain good transferred models. We found that varying the number
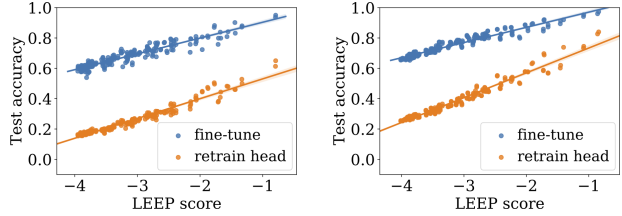


(a) ImageNet → CIFAR100      (b) CIFAR10 → CIFAR100

*Figure 1.* **LEEP scores vs. test accuracies** of two transfer learning algorithms, together with their best fit lines, reported for transferred models on 200 random tasks constructed from CIFAR100 data. Blue and orange points on a vertical line correspond to the same target data set. The source models are (a) ResNet18 pre-trained on ImageNet, and (b) ResNet20 pre-trained on CIFAR10. See Sec. 5.1 for details.

of epochs does not significantly change the results of our experiments, although in principle, fine-tuning the whole network until convergence could decouple its dependence on the source task.

Fig. 1 shows the LEEP scores and the corresponding test accuracies for transferred models on 200 target tasks. Following the correlation analysis by Nguyen et al. (2019) and Tran et al. (2019), we compute the Pearson correlation coefficients and the $p$ values between LEEP scores and test accuracies, which are shown in Table 1 (see the first two rows of each transfer method). From Fig. 1 and Table 1, LEEP scores clearly correlate with test accuracies, with correlation coefficients higher than 0.94 and $p < 0.001$ in all cases. Evidently, LEEP scores are a reliable indicator of the performance of transferred models.

## 5.2. LEEP vs. Transfer Accuracy in Small Data Regime

Transfer learning is often performed in cases where target data sets are small. We next evaluate LEEP in such small data settings. We focus on scenarios where the size of the target data set might be insufficient to train a deep model from scratch, but reasonable for transfer learning to be effective. Particularly, we repeat the experiments of Sec. 5.1 with the additional restriction that target data sets contain exactly five random classes and 50 random examples per class. Thus, the target data sets contain only 10% of the training examples per class, compared to the full CIFAR100 data set.

We further add experiments where target data sets are constructed from the FashionMNIST data set (Xiao et al., 2017). In this case, the target data sets are restricted to contain four random classes and 30 random examples per class (i.e., 0.5% of the training examples per class compared to the full data set). Since FashionMNIST contains grayscale images, we duplicate the grayscale channel three times to be consis-
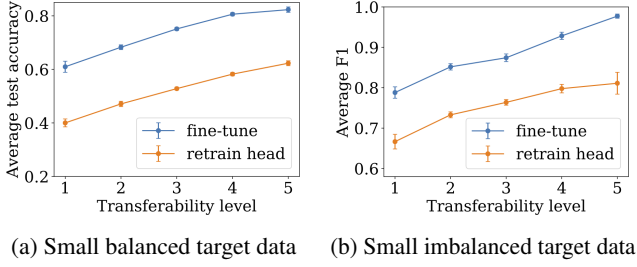
(a) Small balanced target data    (b) Small imbalanced target data

*Figure 2.* **Performance of transferred models on small target data sets in five transferability levels** predicted from LEEP scores. The higher the level, the easier the transfer. The transfer is from a ResNet18 pre-trained on ImageNet to target tasks constructed from CIFAR100. We considered both (a) balanced and (b) imbalanced target data sets. See Sec. 5.2 and 5.3 for details.

tent with the source model's input domain. Because the target data set size is small, we re-train the head classifier by running SGD for only 50 epochs.

Table 1 (see the 3rd to 6th rows of the first two algorithms) shows the correlation coefficients between LEEP scores and test accuracies in these settings. The results indicate a reasonably positive correlations with coefficients greater than 0.5 in most cases, with the exception of fine-tuning from the CIFAR10 pre-trained model. We note that, in general, the number of examples per class in the target data set would affect the effectiveness of LEEP. For instance, if the number of examples is too small, the empirical conditional distribution, and consequently LEEP, would become unreliable.

To factor out the noise when evaluating LEEP scores on small target data sets, we also consider partitioning the scores' range into five equal bins and averaging the test accuracies of tasks in each bin. This allows us to compare the average test accuracies of tasks belonging in five transferability levels. Fig. 2(a) shows a typical result for this case, where higher transferability levels generally imply better accuracies for both transfer methods. Full results are given in the supplementary material. These results testify that LEEP scores can predict accuracies of transferred models, even in small data regimes.

We also evaluate LEEP in the noisy, small data setting, where we repeat the experiment between ImageNet and CIFAR100 but randomly flip the labels in the target data set to a wrong one with probability 0.15. We report the correlation coefficients for this case in Table 1 (see the 7th rows of the first two algorithms). The results also show positive correlations between LEEP scores and test accuracies, although the correlations are weaker than those with correct labels.

### 5.3. LEEP vs. F1 Score on Imbalanced Data

Imbalanced target data sets are commonly encountered in practice (Al-Stouhi & Reddy, 2016; Wang et al., 2017). We

now evaluate LEEP in this setting. Specifically, we also repeat the experiments of Sec. 5.1 and 5.2 where, this time, we restrict target data sets to imbalanced binary classification sets. The two classes of each target data set are drawn randomly. The size of the smaller class is chosen uniformly from 30 to 60, while the other class is five times larger. Because the target data sets are small with only two classes, we only need to re-train the head or fine-tune for 20 epochs.

Table 1 (see the last four rows of the first two algorithms) shows the correlation coefficients between LEEP scores and test F1 scores in these settings. The results also indicate a reasonably positive correlations with coefficients greater than 0.5 in all cases. Fig. 2(b) further shows the average test F1 scores in five transferability levels predicted from LEEP for a typical case, where higher transferability levels imply better F1 scores for both transfer methods. Full results are given in the supplementary material. These results again confirm that LEEP scores can predict the performance of transferred models, even for imbalanced target data sets.

### 5.4. LEEP vs. Accuracy of Meta-Transferred Models

Meta-transfer learning is a framework for learning to adapt from a source task to a target task (Wei et al., 2018b; Sun et al., 2019; Requeima et al., 2019). Next, we show that LEEP can also predict the test accuracies of CNAPs (Requeima et al., 2019), a recently proposed meta-transfer learning method. CNAPs adapt a pre-trained model on the source task to a target task by adding scale and shift parameters to each channel of its convolutional layers. The additional parameters are outputs of adaptation networks, which are trained by meta-learning from several training tasks. When given a target data set, the adaptation networks return the appropriate scale and shift parameters that can augment the source model to make predictions on the target test set.

In our experiment, we follow the original training procedure proposed for CNAPs by Requeima et al. (2019), where the source model is a ResNet18 pre-trained on ImageNet, and the adaptation networks are trained using the Meta-dataset (Triantafillou et al., 2020). We also test CNAPs on 200 random target tasks drawn from CIFAR100 as follows. Each target data set contains five random labels and 50 random examples per class, drawn from the test set of CIFAR100. The remaining 50 test examples of the selected classes are used as the target test set. This testing procedure is consistent with the one used by Requeima et al. (2019), except that in our experiments, we fix the number of classes.

The last row of Table 1 shows the correlation coefficient between LEEP scores and test accuracies of CNAPs. The coefficient is 0.591 with $p < 0.001$, indicating that LEEP scores are a good measure of meta-transferability. Similar to Sec. 5.2 and 5.3, Fig. 3 provides the average test accuracies of tasks in five LEEP score transferability levels. Fig. 3
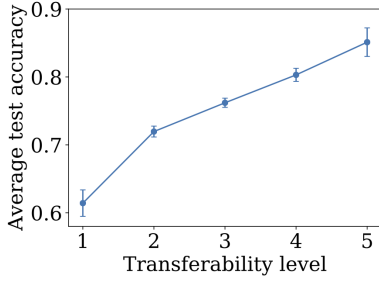
*Figure 3.* **Average test accuracy of CNAPs on tasks in five transferability levels** predicted from LEEP scores. The higher the level, the easier the transfer. See Sec. 5.4 for details.
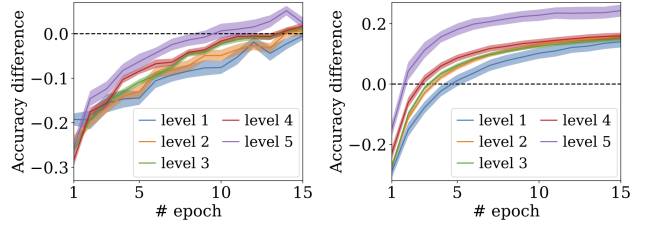


(a) ImageNet → CIFAR100      (b) CIFAR10 → CIFAR100

*Figure 4.* **Convergence of accuracy of fine-tuned models** to the accuracy of *reference* models trained from scratch using only the target data set. The convergence is represented by the accuracy difference between the fine-tuned and the reference models. Each curve is the average over tasks within the same transferability level. The zero lines indicate where the fine-tuned models match the accuracy of the reference models. See Sec. 5.5 for details.

clearly shows that higher transferability levels correspond to better test accuracies for CNAPs.

## 5.5. LEEP vs. Convergence of Fine-tuned Models

For each target task, let us consider a *reference* model: one that is trained from scratch using only the target data set. When it is easier to transfer from a source task to the target task, we often expect the fine-tuned models to converge more quickly and even exceed the performance of this reference model. The experiments in this section show that LEEP scores indeed predict this behavior.

We use the same small data settings defined in Sec. 5.2 and train a reference model for each target task. Reference models are trained using SGD with 100 epochs for target tasks from CIFAR100 and with 40 epochs for those from FashionMNIST. When fine-tuning a model, we track the difference between the fine-tuned model's test accuracy and the *final* test accuracy of the corresponding reference model (i.e., we always compare against the fully trained reference model). Similar to Sec. 5.2, we also consider different transferability levels according to LEEP scores, and average the accuracy differences of all tasks within the same transferability level.

Fig. 4 plots the average accuracy difference curves of five different transferability levels when transferring to CI-FAR100 target tasks. Results for FashionMNIST target tasks are similar and given in the supplementary material. From Fig. 4, on average, fine-tuned models on tasks in higher transferability levels have better convergence speeds (i.e., their curves reach zero faster). Furthermore, these models can outperform the reference models by larger margins (i.e., their curves reach higher values). In all cases, the fine-tuned models match the performance of the reference models using far fewer training epochs. These results confirm the advantage of transfer learning in small data settings, especially between highly transferable tasks, which in turn, can be efficiently predicted using our LEEP scores.

## 5.6. Comparison of LEEP, NCE, and H scores

We compare the LEEP measure with the NCE measure proposed by Tran et al. (2019) and the H score proposed by Bao et al. (2019). Particularly, for all the experimental settings in Sec. 5.1, 5.2, 5.3, and 5.4 above, we compute the NCE score for each transfer from a source model to a target task using the method described in Sec. 3. The computation is efficient since it also requires a single forward pass through the target data set to get the dummy labels. After obtaining the NCE scores, we evaluate their Pearson correlation coefficients and $p$ values with test accuracies (or F1 for imbalanced data) of the three transfer (or meta-transfer) learning algorithms. Similarly, we also compute the correlation coefficients and $p$ values of the H scores between the features returned by the source model and the target labels. The features are obtained by applying the source model to the target inputs.

Table 1 shows the correlation coefficients of NCE and H scores in comparison with those of LEEP scores. When compared with NCE, LEEP scores have equal or better correlations with transfer performance in all except for two cases of the fine-tuning method (see the second and third rows of this method in Table 1). Even in these two cases, LEEP scores are only slightly worse than NCE scores. These comparisons confirm that LEEP scores are better than NCE scores for our transfer settings, with up to 30% improvement in terms of correlation coefficients.

When compared with H scores, LEEP scores give better correlations in 16/23 cases. We note that the performance of H scores is not consistent in all experiments. Particularly, it completely fails to capture the transferability in 11 cases (those marked with asterisks). Even when it successfully captures transferability, it performs worse than LEEP on large target data sets. By comparison, our LEEP scores capture the transferability in all cases.

*Table 1.* **Comparison of Pearson correlation coefficients** of LEEP, NCE (Tran et al., 2019), and H scores (Bao et al., 2019). Correlations are computed with respect to test accuracies (or F1) of three (meta)-transfer learning algorithms in various experimental settings. Correlations marked with asterisks (*) are not statistically significant ($p > 0.05$), while the rest are statistically significant with $p < 0.001$.

| Algorithm | Experiment setting | | | | | Correlation coefficients | | |
|---|---|---|---|---|---|---|---|---|
| | Source | Target | Source model | Properties of target data set | Details in | LEEP | NCE | H |
| Re-train head | CIFAR10 | CIFAR100 | ResNet20 | large, balanced | Sec. 5.1 | **0.982** | **0.982** | 0.831 |
| | ImageNet | CIFAR100 | ResNet18 | large, balanced | Sec. 5.1 | **0.974** | 0.973 | 0.924 |
| | CIFAR10 | CIFAR100 | ResNet20 | small, balanced | Sec. 5.2 | 0.744 | 0.743 | **0.877** |
| | ImageNet | CIFAR100 | ResNet18 | small, balanced | Sec. 5.2 | **0.798** | 0.715 | 0.026* |
| | CIFAR10 | FashionMNIST | ResNet20 | small, balanced | Sec. 5.2 | 0.518 | 0.429 | **0.787** |
| | ImageNet | FashionMNIST | ResNet18 | small, balanced | Sec. 5.2 | **0.631** | 0.622 | 0.005* |
| | ImageNet | CIFAR100 | ResNet18 | small, balanced, noisy | Sec. 5.2 | **0.612** | 0.579 | 0.017* |
| | CIFAR10 | CIFAR100 | ResNet20 | small, imbalanced | Sec. 5.3 | **0.862** | 0.847 | 0.787 |
| | ImageNet | CIFAR100 | ResNet18 | small, imbalanced | Sec. 5.3 | **0.522** | 0.484 | -0.058* |
| | CIFAR10 | FashionMNIST | ResNet20 | small, imbalanced | Sec. 5.3 | 0.704 | 0.688 | **0.822** |
| | ImageNet | FashionMNIST | ResNet18 | small, imbalanced | Sec. 5.3 | **0.645** | 0.624 | 0.059* |
| Fine-tune | CIFAR10 | CIFAR100 | ResNet20 | large, balanced | Sec. 5.1 | **0.967** | **0.967** | 0.787 |
| | ImageNet | CIFAR100 | ResNet18 | large, balanced | Sec. 5.1 | 0.944 | **0.945** | 0.875 |
| | CIFAR10 | CIFAR100 | ResNet20 | small, balanced | Sec. 5.2 | 0.396 | 0.401 | **0.737** |
| | ImageNet | CIFAR100 | ResNet18 | small, balanced | Sec. 5.2 | **0.762** | 0.584 | -0.029* |
| | CIFAR10 | FashionMNIST | ResNet20 | small, balanced | Sec. 5.2 | 0.339 | 0.258 | **0.826** |
| | ImageNet | FashionMNIST | ResNet18 | small, balanced | Sec. 5.2 | **0.609** | 0.578 | 0.018* |
| | ImageNet | CIFAR100 | ResNet18 | small, balanced, noisy | Sec. 5.2 | **0.348** | 0.324 | 0.06* |
| | CIFAR10 | CIFAR100 | ResNet20 | small, imbalanced | Sec. 5.3 | 0.597 | 0.582 | **0.758** |
| | ImageNet | CIFAR100 | ResNet18 | small, imbalanced | Sec. 5.3 | **0.565** | 0.503 | -0.069* |
| | CIFAR10 | FashionMNIST | ResNet20 | small, imbalanced | Sec. 5.3 | 0.603 | 0.589 | **0.904** |
| | ImageNet | FashionMNIST | ResNet18 | small, imbalanced | Sec. 5.3 | **0.507** | 0.425 | 0.056* |
| CNAPS | ImageNet | CIFAR100 | ResNet18 | small, balanced | Sec. 5.4 | **0.591** | 0.310 | 0.025* |

### 5.7. LEEP for Source Model Selection

We evaluate LEEP for the source model selection problem, where we need to select the best source model from 9 candidate models and transfer it to CIFAR100. The candidate models are pre-trained on ImageNet and include ResNet18, ResNet34, ResNet50 (He et al., 2016), MobileNet1.0, MobileNet0.75, MobileNet0.5, MobileNet0.25 (Howard et al., 2017), DarkNet53 (Redmon & Farhadi, 2018), and SENet154 (Hu et al., 2018). Our target data set is the full CIFAR100 training set, while the target test set is the full CIFAR100 test set. We compare our LEEP measure to the NCE (Tran et al., 2019) and H score (Bao et al., 2019) baselines. We also consider ImageNet top-1 accuracy as an additional baseline since previous work (Kornblith et al., 2019) has shown that ImageNet accuracy can predict the performance of transferred models.

Fig. 5 shows the results of this experiment for the head re-training method. From the figure, LEEP scores can predict well the test accuracy of models whose head classifiers are re-trained. In comparison, the other baselines all perform worse than LEEP. For example, NCE fails to predict the performance of MobileNet1.0, while H score and ImageNet

accuracy fail on the SENet154.

We also give results for the fine-tuning method in the supplementary material. Generally, all the considered measures do not predict well the test accuracy of fine-tuned models, especially for ResNet18 and ResNet34 source models. One possible explanation is that the performance of fine-tuned models is sensitive to the architecture and the size of the source networks. However, the transferability measures considered in this section do not take these factors into account.

## 6. Discussions

We proposed LEEP, a novel transferability measure that can efficiently estimate the performance of transfer and meta-transfer learning algorithms before actually executing them. We show both theoretically and empirically that LEEP is an effective transferability measure that is useful in several scenarios. Below are more discussions about our work.

**Source model assumption**. Our work assumes a source model pre-trained on the source task. If, instead, we are given the source data set, we can first train a source model
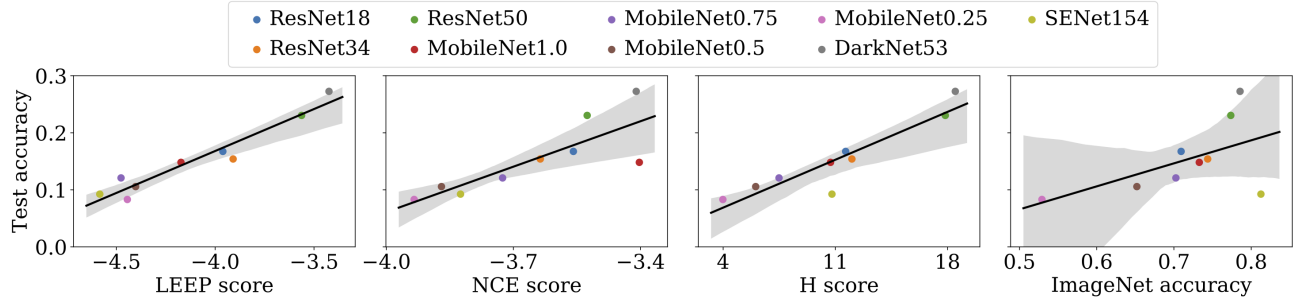
*Figure 5.* **Test accuracy vs. transferability** predicted from LEEP, NCE (Tran et al., 2019), H score (Bao et al., 2019), and ImageNet top-1 accuracy (Kornblith et al., 2019) for 9 candidate source models (see the legend) pre-trained on ImageNet. The transferred models are obtained by re-training the head classifier. See Sec. 5.7 for details.

from the data and then use this model to compute the LEEP scores. The LEEP scores would depend on the architectural choice and thus the performance of the source model. This is expected and has been pointed out in previous work (Kornblith et al., 2019; Tran et al., 2019).

**Rationale for LEEP and possible extensions**. The design of LEEP was aimed at being *algorithm independence*, i.e., it should work for many different transfer learning algorithms. In fact, the performance of transfer learning algorithms depends partially on the relationship between the source model and the target data set, and thus LEEP tries to quantify such relationship to achieve this independence. We note that LEEP was not intended to be a transfer learning algorithm although the EEP can be viewed as a transfer learning algorithm that was designed for simplicity and speed instead of accuracy. Transfer learning should be performed by a proper algorithm, e.g., by re-training the head classifier or fine-tuning the model parameters.

Although it might seem surprising that LEEP works well in our experiments without explicitly using the feature distributions, we note that LEEP in fact takes into account the feature distributions *indirectly*. Specifically, given a pre-trained source network, the output label distribution is a linear transformation of the features (i.e., output of the penultimate layer) followed by the Softmax. Thus, the dummy source label distribution indirectly contains information about the input features.

We can extend LEEP to include the learned features by transforming the feature vector into a probability distribution directly using a Softmax, and then computing LEEP with this new dummy distribution. In this case, the support of the dummy distribution has the same length as that of the feature vector and would have a different interpretation than ours. This would be a direction for future work.

**Effects of heterogeneous source and target tasks**. LEEP can be applied when the source and target tasks have differ-

ent label semantics or when their input spaces are heterogeneous. In these cases, the source model would be more uncertain about the inputs from the target data set, leading to a more uniform dummy label distribution. By definition, LEEP scores will be smaller and thus indicating a harder transfer for these cases.

**Applications of LEEP**. Our experiments, reported in Sec. 5, showed that LEEP is applicable in diverse scenarios. In general, LEEP scores can be used to efficiently select highly transferable pairs of source and target tasks, yielding high transfer accuracy and good convergence speeds. This ability can support source model selection in transfer/meta-transfer learning scenarios, where we need to select a source model among several others in order to optimize for best performance on a target task.

Aside from transfer and meta-transfer learning, our LEEP scores are potentially useful for continual learning (Zenke et al., 2017; Swaroop et al., 2019), multi-task learning (Misra et al., 2016; Standley et al., 2019), and feature selection (Yosinski et al., 2014). For instance, LEEP scores can be used to estimate the hardness of task sequences, thereby helping to analyze properties of continual learning algorithms (Nguyen et al., 2019). For multi-task learning, LEEP scores can be used for selecting groups of tasks for joint training (Standley et al., 2019). Finally, LEEP scores could be useful for hyperparameter transfer learning and Bayesian optimization (Perrone et al., 2018). These are promising research directions that we leave to future work.

# References

Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C. C., Soatto, S., and Perona, P. Task2vec: Task embedding for meta-learning. In *International Conference on Computer Vision*, pp. 6430–6439, 2019.

Agrawal, P., Girshick, R., and Malik, J. Analyzing the performance of multilayer neural networks for object

recognition. In *European Conference on Computer Vision*, pp. 329–344, 2014.

Al-Stouhi, S. and Reddy, C. K. Transfer learning for class imbalance problems with inadequate data. *Knowledge and information systems*, 48(1):201–228, 2016.

Ammar, H. B., Eaton, E., Taylor, M. E., Mocanu, D. C., Driessens, K., Weiss, G., and Tuyls, K. An automated measure of MDP similarity for transfer in reinforcement learning. In *AAAI Conference on Artificial Intelligence Workshops*, 2014.

Azizzadenesheli, K., Liu, A., Yang, F., and Anandkumar, A. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019.

Bao, Y., Li, Y., Huang, S.-L., Zhang, L., Zheng, L., Zamir, A., and Guibas, L. An information-theoretic approach to transferability in task transfer learning. In *IEEE International Conference on Image Processing*, pp. 2309–2313, 2019.

Ben-David, S. and Schuller, R. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pp. 567–580. Springer, 2003.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 137–144, 2007.

Bhattacharjee, B., Codella, N., Kender, J. R., Huo, S., Watson, P., Glass, M. R., Dube, P., Hill, M., and Belgodere, B. P2L: Predicting transfer learning for images and semantic relations. *arXiv:1908.07630*, 2019.

Bottou, L. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

Cao, B., Pan, S. J., Zhang, Y., Yeung, D.-Y., and Yang, Q. Adaptive transfer learning. In *AAAI Conference on Artificial Intelligence*, 2010.

Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., and Zhang, Z. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274*, 2015.

Dhillon, G. S., Chaudhari, P., Ravichandran, A., and Soatto, S. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, pp. 647–655, 2014.

Eaton, E., Lane, T., et al. Modeling transfer relationships between learning tasks for improved inductive transfer. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 317–332, 2008.

Edwards, H. and Storkey, A. Towards a neural statistician. In *International Conference on Learning Representations*, 2017.

Gebelein, H. Das statistische problem der korrelation als variations-und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 21(6):364–379, 1941.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

Guo, J., He, H., He, T., Lausen, L., Li, M., Lin, H., Shi, X., Wang, C., Xie, J., Zha, S., et al. GluonCV and GluonNLP: Deep learning in computer vision and natural language processing. *arXiv:1907.04433*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hirschfeld, H. O. A connection between correlation and contingency. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pp. 520–524, 1935.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

Jomaa, H. S., Grabocka, J., and Schmidt-Thieme, L. Dataset2vec: Learning dataset meta-features. *arXiv:1905.11063*, 2019.

Kifer, D., Ben-David, S., and Gehrke, J. Detecting change in data streams. In *International Conference on Very Large Data Bases*, volume 4, pp. 180–191, 2004.

Kingma, D. P. and Welling, M. Stochastic gradient vb and the variational auto-encoder. In *International Conference on Learning Representations*, volume 19, 2014.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2661–2671, 2019.

Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*. Master's thesis, University of Toronto, 2009.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In *Annual Conference on Learning Theory*, 2009.

Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. Cross-stitch networks for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3994–4003, 2016.

Nguyen, C. V., Achille, A., Lam, M., Hassner, T., Mahadevan, V., and Soatto, S. Toward understanding catastrophic forgetting in continual learning. *arXiv:1908.01091*, 2019.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717–1724, 2014.

Perrone, V., Jenatton, R., Seeger, M. W., and Archambeau, C. Scalable hyperparameter transfer learning. In *Advances in Neural Information Processing Systems*, pp. 6845–6855, 2018.

Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813, 2014.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018.

Rényi, A. On measures of dependence. *Acta Mathematica Hungarica*, 10(3-4):441–451, 1959.

Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., and Turner, R. E. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems*, pp. 7957–7968, 2019.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Li, F.-F. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Sinapov, J., Narvekar, S., Leonetti, M., and Stone, P. Learning inter-task transferability in the absence of target task samples. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 725–733, 2015.

Standley, T., Zamir, A. R., Chen, D., Guibas, L., Malik, J., and Savarese, S. Which tasks should be learned together in multi-task learning? *arXiv:1905.07553*, 2019.

Sun, B., Feng, J., and Saenko, K. Return of frustratingly easy domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2016.

Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. Meta-transfer learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

Swaroop, S., Nguyen, C. V., Bui, T. D., and Turner, R. E. Improving and understanding variational continual learning. *arXiv:1905.02099*, 2019.

Tran, A. T., Nguyen, C. V., and Hassner, T. Transferability and hardness of supervised classification tasks. In *International Conference on Computer Vision*, pp. 1395–1405, 2019.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., and Larochelle, H. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

Wang, J., Chen, Y., Hao, S., Feng, W., and Shen, Z. Balanced distribution adaptation for transfer learning. In *IEEE International Conference on Data Mining*, pp. 1129–1134, 2017.

Wei, P., Sagarna, R., Ke, Y., and Ong, Y. S. Uncluttered domain sub-similarity modeling for transfer regression. In *IEEE International Conference on Data Mining*, pp. 1314–1319, 2018a.

Wei, Y., Zhang, Y., Huang, J., and Yang, Q. Transfer learning via learning to transfer. In *International Conference on Machine Learning*, pp. 5085–5094, 2018b.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

Whatmough, P. N., Zhou, C., Hansen, P., Venkataramanaiah, S. K., Seo, J.-s., and Mattina, M. FixyNN: Efficient hardware for mobile computer vision via transfer learning. In *Conference on Systems and Machine Learning*, 2019.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

Yang, Q., Zhang, Y., Dai, W., and Pan, S. J. *Transfer learning*. Cambridge University Press, 2020.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3712–3722, 2018.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pp. 818–833, 2014.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995, 2017.