Simple Transferability Estimation for Regression Tasks

Abstract

We consider transferability estimation, the problem of estimating how well deep learning models transfer from a source to a target task. We focus on regression tasks, which received little previous attention, and propose two simple and computationally efficient approaches that estimate transferability based on the negative regularized mean squared error of a linear regression model. We prove novel theoretical results connecting our approaches to the actual transferability of the optimal target models obtained from the transfer learning process. Despite their simplicity, our approaches significantly outperform existing state-of-the-art regression transferability estimators in both accuracy and efficiency. On two large-scale keypoint regression benchmarks, our approaches yield 9% to 26% better results on average while being at least 27% faster than previous state-of-the-art methods.

1 INTRODUCTION

Transferability estimation [Bao et al., 2019, Tran et al., 2019, Nguyen et al., 2020] aims to develop computationally efficient metrics to predict the effectiveness of transferring a deep learning model from a source to a target task. This problem has recently gained attention as a means for model and task selection [Bao et al., 2019, Tran et al., 2019, Nguyen et al., 2020, Bolya et al., 2021, You et al., 2021] that can potentially improve the performance and reduce the cost of transfer learning, especially for expensive deep learning models. In recent years, new transferability estimators were also developed and used in applications such as checkpoint ranking [Huang et al., 2021, Li et al., 2021] and few-shot learning [Tong et al., 2021].

Nearly all existing methods consider only the transferability between classification tasks [Bao et al., 2019, Tran et al., 2019, Nguyen et al., 2020, Deshpande et al., 2021, Li et al., 2021, Tan et al., 2021, Huang et al., 2022], with very few designed for regression [You et al., 2021, Huang et al., 2022], despite the importance of regression problems in a wide range of applications such as landmark detection [Fard et al., 2021, Poster et al., 2021], object detection and localization [Cai et al., 2020, Bu et al., 2021], pose estimation [Schwarz et al., 2015, Doersch and Zisserman, 2019], or image generation [Ramesh et al., 2021, Razavi et al., 2019]. Moreover, those few methods are often a byproduct of a classification transferability estimator and were never tested against regression transferability estimation baselines.

In this paper, we explicitly consider transferability estimation for regression tasks and formulate a novel definition for this problem. Our formulation is based on the practical usage of transferability estimation: to compare the actual transferability between different tasks [Bao et al., 2019, Tran et al., 2019, Nguyen et al., 2020, You et al., 2021]. We then propose two *simple, efficient, and theoretically grounded* approaches for this problem that estimate transferability using the negative regularized mean squared error (MSE) of a linear regression model computed from the source and target training sets. The first approach, *Linear MSE*, uses the linear regression model between features extracted from the source model (a model trained on the source task) and true labels of the target training set. The second approach, *Label MSE*, estimates transferability by regressing between the *dummy* labels, obtained from the source model, and true labels of the target data. In special cases where the source and target data share the inputs, the Label MSE estimators can be computed even more efficiently from the true labels without a source model.

In addition to their simplicity, we show our transferability estimators to have theoretical properties relating them to the actual transferability of the transferred target model. In particular, we prove that the transferability of the target model obtained from transfer learning is lower bounded by the Label MSE minus a complexity term, which depends on the target dataset size and the model architecture. Similar

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

theoretical results can also be proven for the case where the source and target tasks share the inputs.

We conduct extensive experiments on two real-world keypoint detection datasets, CUB-200-2011 [Wah et al., 2011] and OpenMonkey [Yao et al., 2021], as well as the dSprites shape regression dataset [Matthey et al., 2017] to show the advantages of our approaches. The results clearly demonstrate that despite their simplicity, our approaches outperform recently published, state-of-the-art (SotA) regression transferability estimators, such as LogME [You et al., 2021] and TransRate [Huang et al., 2022], in both effectiveness and efficiency. In particular, our approaches can improve SotA results from 9% to 26% on average, while being at least 27% faster.

Summary of contributions. (1) We formulate a new definition for the transferability estimation problem that can be used for comparing the actual transferability (§3). (2) We propose Linear MSE and Label MSE, two simple yet effective transferability estimators for regression tasks (§4). (3) We prove novel theoretical results for these estimators to connect them with the actual task transferability (§5). (4) We rigorously test our approaches in various settings and challenging benchmarks, showing their advantages compared to SotA regression transferability methods (§6).

2 RELATED WORK

Our paper is one of the recent attempts to develop efficient and effective transferability estimators for deep transfer learning [Bao et al., 2019, Tran et al., 2019, Nguyen et al., 2020, Deshpande et al., 2021, Li et al., 2021, Tan et al., 2021, You et al., 2021, Huang et al., 2022, Nguyen et al., 2022]. Most of the existing work for transferability estimation focuses on classification [Bao et al., 2019, Tran et al., 2019, Nguyen et al., 2020, Deshpande et al., 2021, Li et al., 2021, Tan et al., 2021, Nguyen et al., 2022], while we are only aware of two methods developed for regression [You et al., 2021, Huang et al., 2022]. One method, called LogME [You et al., 2021], takes a Bayesian approach and uses the maximum log evidence of the target data as the transferability estimator. While this method can be sped up using matrix decomposition, its scalability is still limited since the required memory is large. In contrast, our proposed approaches are simpler, faster, and more effective. We also provide novel theoretical properties for our methods that were not available for LogME. Another approach for transferability estimation between regression tasks, called TransRate [Huang et al., 2022], is to divide the real-valued outputs into different bins and apply a classification transferability estimator. In our experiments, we will show that this approach is less accurate than both LogME and our approaches.

Transferability can also be inferred from a task taxonomy [Zamir et al., 2018, Dwivedi and Roig, 2019, Dwivedi et al., 2020] or a task space representation [Achille et al., 2019], which embeds tasks as vectors on a vector space. A popular task taxonomy, Taskonomy [Zamir et al., 2018], exploits the underlying structure of visual tasks by computing a task affinity matrix that can be used for estimating transferability. Constructing the Taskonomy requires training a small classification head, which resembles the training of the regularized linear regression models in our approaches. However, they investigate the global taxonomy of classification tasks, while our paper studies regression tasks with a focus on estimating their transferability efficiently.

Our paper is also related to transfer learning for deep models [Tan et al., 2018], which has been successful in real-world regression problems such as object detection and localization [Cai et al., 2020, Bu et al., 2021], landmark detection [Fard et al., 2021, Poster et al., 2021], or pose estimation [Schwarz et al., 2015, Doersch and Zisserman, 2019]. Several previous works have investigated theoretical bounds for transfer learning [Ben-David and Schuller, 2003, Blitzer et al., 2007, Mansour et al., 2009, Azizzadenesheli et al., 2019, Wang et al., 2019, Tripuraneni et al., 2020]; however, these bounds are hard to compute in practice and thus unsuitable for transferability estimation. Some previous transferability estimators have theoretical bounds on the empirical loss of the transferred model [Tran et al., 2019, Nguyen et al., 2020], but these bounds were for classification and did not relate directly to transferability. Our bounds, on the other hand, focus on regression and connect our approaches directly to the notion of transferability.

3 TRANSFERABILITY BETWEEN REGRESSION TASKS

In this section, we describe the transfer learning setting that will be used in our subsequent analysis. We then propose a definition of transferability for regression tasks and a new formulation for the transferability estimation problem.

3.1 TRANSFER LEARNING FOR REGRESSION

Consider a source training set $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and a target training set $\mathcal{D}_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ consisting of n_s and n_t examples respectively, where $x_i^s, x_i^t \in \mathbb{R}^d$ are d-dimensional input vectors, $y_i^s \in \mathbb{R}^{d_s}$ is a d_s-dimensional source label vector, and $y_i^t \in \mathbb{R}^{d_t}$ is a d_t-dimensional target label vector. Here we allow multi-output regression tasks (with $d_s, d_t \geq 1$) where the source and target labels may have different dimensions ($d_s \neq d_t$). In the simplest case, the source and target tasks are both single-output regression tasks where $d_s = d_t = 1$.

In this paper, we will refer to a model (such as w, w^*, h, h^*, k, or k^*) and its parameters interchangeably. Using the source dataset $\mathcal{D}_s$, we train a deep learning model (w^*, h^*) consisting of an optimal feature extractor w^* and an optimal regression head h^* that minimizes the empirical MSE loss:

$$w^*, h^* = \operatorname{argmin}_{w,h} \mathcal{L}(w, h; \mathcal{D}_s), \tag{1}$$

where $w : \mathbb{R}^d \to \mathbb{R}^{d_r}$ is a feature extractor network

that transforms a d-dimensional input vector into a d_r-dimensional feature vector, $h : \mathbb{R}^{d_r} \to \mathbb{R}^{d_s}$ is a source regression head network that transforms a d_r-dimensional feature vector into a d_s-dimensional output vector, and $\mathcal{L}(w, h; \mathcal{D}_s)$ is the empirical MSE loss of the whole model (w, h) on the dataset $\mathcal{D}_s$:

$$\mathcal{L}(w, h; \mathcal{D}_s) = \frac{1}{n_s} \sum_{i=1}^{n_s} \|y_i^s - h(w(x_i^s))\|^2, \qquad (2)$$

with $\|\cdot\|$ being the ℓ_2 norm. In practice, we usually consider a source model (e.g., a ResNet [He et al., 2016]) as a whole and use its first l layers from the input (for some chosen number l) as the feature extractor w. The regression head h is the remaining part of the model from the l-th layer to the output layer, and the prediction for any input x is $h(w(x))$.

After training the optimal source model (w^*, h^*), we perform transfer learning to the target task by freezing the optimal feature extractor w^* and re-training a new regression head k^* using the target dataset $\mathcal{D}_t$, also by minimizing the empirical MSE loss:

$$\begin{aligned} k^* &= \text{argmin}_k \, \mathcal{L}(w^*, k; \mathcal{D}_t) \\ &= \text{argmin}_k \{ \frac{1}{n_t} \sum_{i=1}^{n_t} \|y_i^t - k(w^*(x_i^t))\|^2 \}, \end{aligned} \qquad (3)$$

where $k : \mathbb{R}^{d_r} \to \mathbb{R}^{d_t}$ is a target regression head network that may have a different architecture than that of h. In general, the regression heads h and k may contain multiple layers and are not necessarily linear.

This transfer learning algorithm, usually called *head re-training*, has been widely used for deep learning models [Donahue et al., 2014, Oquab et al., 2014, Sharif Razavian et al., 2014, Whatmough et al., 2019] and will be used for our theoretical analysis. In practice and in our experiments, we also consider another transfer learning algorithm, widely known as *fine-tuning*, where we fine-tune the trained feature extractor w^* on the target set, and then train a new target regression head k^* with this fine-tuned feature extractor [Agrawal et al., 2014, Girshick et al., 2014, Chatfield et al., 2014, Dhillon et al., 2020].

3.2 TRANSFERABILITY ESTIMATION

As our first contribution, we propose a definition of transferability for regression tasks and a new formulation for the transferability estimation problem. For this purpose, we make the standard assumption that the source data $\mathcal{D}_s$ and the target data $\mathcal{D}_t$ are drawn iid from the true but unknown distributions $\mathbb{P}(X^s, Y^s)$ and $\mathbb{P}(X^t, Y^t)$; that is, $(x_i^s, y_i^s) \overset{\text{iid}}{\sim} \mathbb{P}(X^s, Y^s)$ and $(x_i^t, y_i^t) \overset{\text{iid}}{\sim} \mathbb{P}(X^t, Y^t)$. In this case, the true source data distribution $\mathbb{P}(X^s, Y^s)$ can represent the source task T_s and the true target data distribution $\mathbb{P}(X^t, Y^t)$ can represent the target task T_t.

Let (w^*, k^*) be a target model obtained using a transfer learning algorithm. We define the transferability between two regression tasks T_s and T_t as the expected negative ℓ_2 loss of this transferred model (w^*, k^*) on a random labeled example drawn from the true target distribution $\mathbb{P}(X^t, Y^t)$, as in Definition 3.1 below. From this definition, the lower the loss of (w^*, k^*), the higher the transferability.

Definition 3.1. *The **transferability** between a source task T_s and a target task T_t is defined as:* $\text{Tr}(T_s, T_t) := \mathbb{E}_{(x^t, y^t) \sim \mathbb{P}(X^t, Y^t)} \left\{ -\|y^t - k^*(w^*(x^t))\|^2 \right\}$.

In previous work [Tran et al., 2019, Huang et al., 2022], transferability estimation was defined as estimating $\text{Tr}(T_s, T_t)$ from their training sets $(\mathcal{D}_s, \mathcal{D}_t)$; that is, to derive a real-valued metric $\mathcal{T}(\mathcal{D}_s, \mathcal{D}_t) \in \mathbb{R}$ such that $\mathcal{T}(\mathcal{D}_s, \mathcal{D}_t) \approx \text{Tr}(T_s, T_t)$. However, in most applications of transferability estimation such as task selection [Tran et al., 2019, Huang et al., 2022, You et al., 2021] or model ranking [Huang et al., 2021, Li et al., 2021], an accurate approximation of $\text{Tr}(T_s, T_t)$ is usually not required since $\mathcal{T}(\mathcal{D}_s, \mathcal{D}_t)$ is only used for comparing tasks or models. Thus, we propose below an *alternative definition* for this problem that better aligns with its practical usage.

Definition 3.2. ***Transferability estimation** aims to find a computationally efficient real-valued metric $\mathcal{T}(\mathcal{D}_s, \mathcal{D}_t) \in \mathbb{R}$ for any pair of training datasets $(\mathcal{D}_s, \mathcal{D}_t)$ such that: $\mathcal{T}(\mathcal{D}_s, \mathcal{D}_t) \leq \mathcal{T}(\mathcal{D}'_s, \mathcal{D}'_t)$ if and only if $\text{Tr}(T_s, T_t) \leq \text{Tr}(T'_s, T'_t)$, where T_s, T_t, T'_s, T'_t are the tasks corresponding with the training datasets $\mathcal{D}_s, \mathcal{D}_t, \mathcal{D}'_s, \mathcal{D}'_t$.*

In this new definition, a transferability estimator $\mathcal{T}(\mathcal{D}_s, \mathcal{D}_t)$ is a function of $(\mathcal{D}_s, \mathcal{D}_t)$ that can be used for comparing transferability. It does *not* need to be an approximation of $\text{Tr}(T_s, T_t)$. This definition is a generalization of previous definitions [Nguyen et al., 2020, Huang et al., 2022] and can be used for source task selection (when $T_t = T'_t$ and $\mathcal{D}_t = \mathcal{D}'_t$) [You et al., 2021] as well as target task selection ($T_s = T'_s$ and $\mathcal{D}_s = \mathcal{D}'_s$) [Nguyen et al., 2020]. It is consistent with the usage of transferability estimators and the way they are evaluated in the literature (e.g., by correlation analysis) [Tran et al., 2019, Nguyen et al., 2020, You et al., 2021, Huang et al., 2022]. In cases where we only have a pre-trained source model $\mathcal{M}_s$ instead of the source training data $\mathcal{D}_s$, the transferability estimator can be defined as $\mathcal{T}(\mathcal{M}_s, \mathcal{D}_t)$, which is a function of $(\mathcal{M}_s, \mathcal{D}_t)$.

4 SIMPLE TRANSFERABILITY ESTIMATORS FOR REGRESSION

In theory, we can use $-\mathcal{L}(w^*, k^*; \mathcal{D}_t)$, the negative MSE of the transferred target model (w^*, k^*), as a transferability estimator, since it is an empirical estimation of $\text{Tr}(T_s, T_t)$ using the dataset $\mathcal{D}_t$. However, this method requires us to run the actual transfer learning process, which could be expensive if the network architecture of the target regression heads (e.g., k and k^*) is deep and complex. This violates a crucial requirement for a transferability estimator in Definition 3.2: the estimator must be *computationally efficient*

since it will be computed several times for task comparison. In this section, we propose two simple regression transferability estimators to address this problem.

4.1 LINEAR MSE ESTIMATOR

To reduce the cost of computing $\mathcal{L}(w^*, k^*; \mathcal{D}_t)$, a simple idea is to approximate it with an ℓ_2-regularized linear regression (Ridge regression) head. This leads to our first simple transferability estimator, Linear MSE, which is defined as the negative regularized MSE of this Ridge regression head. In this definition, $\|\cdot\|_F$ is the Frobenius norm.

Definition 4.1. *The **Linear MSE** transferability estimator with a regularization parameter $\lambda \geq 0$ is: $\mathcal{T}_\lambda^{\text{lin}}(\mathcal{D}_s, \mathcal{D}_t) := -\min_{A,B} \{\frac{1}{n_t} \sum_{i=1}^{n_t} \|y_i^t - Aw^*(x_i^t) - B\|^2 + \lambda \|A\|_F^2\}$, where $A \in \mathbb{R}^{d_r \times d_t}$ is a $d_r \times d_t$ real-valued matrix and $B \in \mathbb{R}^{d_t}$ is a d_t-dimensional real-valued vector.*

Here we add a regularizer to avoid overfitting when the target dataset $\mathcal{D}_t$ is small. Previous work such as LogME [You et al., 2021] proposed to prevent overfitting by taking a Bayesian approach, which is more complicated and expensive. We will show empirically in our experiments (Section 6.3) that this simple regularization approach can tackle the issue more effectively and efficiently.

Given a pre-trained feature extractor w^* and a target set $\mathcal{D}_t$, we can compute $\mathcal{T}_\lambda^{\text{lin}}(\mathcal{D}_s, \mathcal{D}_t)$ efficiently using the closed form solution for Ridge regression or using second-order optimization [Bishop, 2006]. If the target regression head k^* is a linear regression model, $\mathcal{T}_0^{\text{lin}}(\mathcal{D}_s, \mathcal{D}_t)$ with $\lambda = 0$ is the negative MSE of the transferred target model (w^*, k^*) on $\mathcal{D}_t$. If k^* has more than one layer with a non-linear activation, $\mathcal{T}_\lambda^{\text{lin}}(\mathcal{D}_s, \mathcal{D}_t)$ can be regarded as using a regularized linear model to approximate this non-linear head.

4.2 LABEL MSE ESTIMATOR

Although the Linear MSE transferability score above can be computed efficiently, this computation may still be expensive if the feature vectors $w^*(x_i^t)$ are high-dimensional. To further reduce the costs, we propose another transferability estimator, Label MSE, which replaces $w^*(x_i^t)$ by the "dummy" source label $z_i = h^*(w^*(x_i^t))$. Using dummy labels from the pre-trained source model (w^*, h^*) is a technique previously used to compute the LEEP transferability score for classification [Nguyen et al., 2020]. We define this estimator below.

Definition 4.2. *The **Label MSE** transferability estimator with a regularization parameter $\lambda \geq 0$ is: $\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) := -\min_{A,B} \{\frac{1}{n_t} \sum_{i=1}^{n_t} \|y_i^t - Az_i - B\|^2 + \lambda \|A\|_F^2\}$, where $A \in \mathbb{R}^{d_s \times d_t}$ is a $d_s \times d_t$ real-valued matrix, $B \in \mathbb{R}^{d_t}$ is a d_t-dimensional real-valued vector, and $z_i = h^*(w^*(x_i^t))$.*

In practice, since the size of z_i is usually much smaller than that of $w^*(x_i^t)$ (i.e., $d_s \ll d_r$), computing the Label MSE is usually faster than computing the Linear MSE.

Special case with shared inputs. When the source and target datasets have the same inputs, i.e., $\mathcal{D}_s = \{(x_i, y_i^s)\}_{i=1}^n$ and $\mathcal{D}_t = \{(x_i, y_i^t)\}_{i=1}^n$, we can compute the Label MSE even faster using only the true labels. Particularly, we can consider the following version of the Label MSE.

Definition 4.3. *The **Shared Inputs Label MSE** transferability estimator with a regularization parameter $\lambda \geq 0$ is: $\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) := -\min_{A,B} \{\frac{1}{n} \sum_{i=1}^{n} \|y_i^t - Ay_i^s - B\|^2 + \lambda \|A\|_F^2\}$, where $A \in \mathbb{R}^{d_s \times d_t}$ and $B \in \mathbb{R}^{d_t}$.*

In this definition, the Shared Inputs Label MSE is computed by training a Ridge regression model directly from the true label pairs (y_i^s, y_i^t), which is *less expensive* than the original Label MSE since we do not need to train the source model (w^*, h^*) or compute the dummy labels.

While the Linear MSE can be interpreted as a linear approximation to $-\mathcal{L}(w^*, k^*; \mathcal{D}_t)$, properties of the Label MSE and Shared Inputs Label MSE are not well understood. In the next section, we shall prove novel theoretical properties for these estimators.

5 THEORETICAL PROPERTIES

We now prove some theoretical properties for the Label MSE with ReLU feed-forward neural networks. These properties are in the form of generalization bounds relating $\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)$ with the transferability $\text{Tr}(T_s, T_t)$. Throughout this section, we assume the space of all target regression heads k, which may have more than one layer, is a superset of all the linear regression models. This assumption is generally true for ReLU networks [Arora et al., 2018].

First, we show in Lemma 5.1 below a relationship between the negative MSE loss $-\mathcal{L}(w^*, k^*; \mathcal{D}_t)$ of (w^*, k^*) and the Label MSE. This lemma states that the negative MSE loss $-\mathcal{L}(w^*, k^*; \mathcal{D}_t)$ upper bounds the Label MSE. The proof for this lemma is in the supplementary.

Lemma 5.1. *For any $\lambda \geq 0$, we have: $\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) \leq -\mathcal{L}(w^*, k^*; \mathcal{D}_t)$.*

Using this lemma, we can prove our main theoretical result, Theorem 5.2, below. In this theorem, L is the number of layers of the ReLU feed-forward neural network (w^*, k^*), and we assume the number of hidden nodes and parameters in each layer are upper bounded by H and $M \geq 1$ respectively. Without loss of generality, we also assume all input and output data are upper bounded by 1 in ℓ_∞-norm. This assumption can easily be satisfied by a pre-processing step that scales them to $[0, 1]$ in ℓ_∞-norm.

Theorem 5.2. *For any $\lambda \geq 0$ and $\delta > 0$, with probability at least $1 - \delta$, we have: $\text{Tr}(T_s, T_t) \geq \mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) - C(d, d_t, M, H, L, \delta)/\sqrt{n_t}$, where $C(d, d_t, M, H, L, \delta) = 16M^{2L+2}H^{2L}[d_t^2 d\sqrt{L + 1 + \ln d} + d_t d^2 \sqrt{2\ln(4/\delta)}]$.*

The proof for this theorem is in the supplementary. The theorem shows that the transferability $\text{Tr}(T_s, T_t)$ is lower

Table 1: **Correlation coefficients when transferring from OpenMonkey to CUB-200-2011**. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods. Detailed correlation plots are in the supplementary. Our estimators improve up to 25.9% in comparison with SotA (LogME) while being 9.4% better on average.

Transfer setting	Label-based method				Feature-based method			
	LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
Head re-training	0.958	0.165	**0.992***	**0.992***	0.969	0.121	0.982	0.988*
Half fine-tuning	0.706	0.392	**0.882***	**0.882***	0.870	0.304	0.865	0.872*
Full fine-tuning	0.691	0.410	**0.870***	**0.870***	0.861*	0.311	0.854	0.861*

bounded by the Label MSE $\mathcal{T}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)$ minus a complexity term $C(d, d_t, M, H, L, \delta)/\sqrt{n_t}$ that depends on the target dataset (specifically, the input and output dimensions, as well as the dataset size) and the architecture of the target network. When this complexity term is small (e.g., when n_t is large enough), the bound in Theorem 5.2 will be tighter. In this case, a higher Label MSE score will likely lead to better transferability.

Shared inputs case. We can also prove similar bounds for the Shared Inputs Label MSE $\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)$. Denote $A_\lambda^*, B_\lambda^* := \text{argmin}_{A,B} \{\frac{1}{n}\sum_i \|y_i^t - Ay_i^s - B\|^2 + \lambda\|A\|_F^2\}$. We first show the following lemma relating $\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)$ and the losses of the source and target models.

Lemma 5.3. *For any $\lambda \geq 0$, we have: $\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) \leq -\mathcal{L}(w^*, k^*; \mathcal{D}_t)/2 + \|A_\lambda^*\|_F^2 \mathcal{L}(w^*, h^*; \mathcal{D}_s)$.*

Using this lemma, we can prove the following theorem for this shared inputs setting. The proofs for these results are in the supplementary.

Theorem 5.4. *For any $\lambda \geq 0$ and $\delta > 0$, with probability at least $1 - \delta$, we have: $\text{Tr}(T_s, T_t) \geq 2\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t) - 2\|A_\lambda^*\|_F^2 \mathcal{L}(w^*, h^*; \mathcal{D}_s) - C(d, d_t, M, H, L, \delta)/\sqrt{n}$.*

From the theorem, $\widehat{\mathcal{T}}_\lambda^{\text{lab}}(\mathcal{D}_s, \mathcal{D}_t)$ can indirectly tell us information about the transferability $\text{Tr}(T_s, T_t)$ without actually training w^*, h^*, and k^*. This bound becomes tighter when n is large or $\mathcal{L}(w^*, h^*; \mathcal{D}_s)$ is small (e.g., when the source model is expressive enough to fit the source data). An experiment to investigate the usefulness of our theoretical bounds in this section is available in the supplementary.

6 EXPERIMENTS

In this section, we conduct experiments to evaluate our approaches on the keypoint (or landmark) regression tasks using the following two large-scale public datasets:

• **CUB-200-2011** [Wah et al., 2011]. This dataset contains 11,788 bird images with 15 labeled keypoints indicating 15 different parts of a bird body. We use 9,788 images for training and 2,000 images for testing. Since the annotations for occluded keypoints are highly inaccurate, we remove all occluded keypoints during the training for both source and target tasks.

• **OpenMonkey** [Yao et al., 2021]. This is a benchmark for the non-human pose tracking problem. It offers over 100,000 monkey images in natural contexts, annotated with 17 body landmarks. We use the original train-test split, which contains 66,917 training images and 22,306 testing images.

In our experiments, we use ResNet34 [He et al., 2016] as the backbone since it provides good performance as a source model. Following previous work [Tran et al., 2019, Nguyen et al., 2020, Huang et al., 2022, Nguyen et al., 2022], we investigate how well our transferability estimators correlate (using Pearson correlation) with the *negative test MSE* of the target model obtained from actual transfer learning. This correlation analysis is a good method to measure how well transferability estimators satisfy our Definition 3.2.

We consider three standard transfer learning algorithms: (1) **head re-training** [Donahue et al., 2014, Sharif Razavian et al., 2014]: We fix all layers of the source model up until the penultimate layer and re-train the last fully-connected (FC) layer using the target training set; (2) **half fine-tuning** [Donahue et al., 2014, Sharif Razavian et al., 2014]: We fine-tune the last convolutional block and all the FC layers of the source model, while keeping all other layers fixed; and (3) **full fine-tuning** [Agrawal et al., 2014, Girshick et al., 2014]: We fine-tune the whole source model using the target training set. Among these settings, head re-training resembles the transfer scenario in Section 3.1, while half and full fine-tuning are more commonly used in practice. For half fine-tuning, around half of the parameters in the network will be fine-tuned ($\sim$13M parameters). More details of our experiment settings are in the supplementary.

We compare our transferability estimators, Linear MSE and Label MSE, with two recent SotA baselines for regression: LogME [You et al., 2021] and TransRate [Huang et al., 2022]. For our methods, we consider $\lambda = 0$ (named **LinMSE0** and **LabMSE0**) for the estimators without regularization, and $\lambda = 1$ (named **LinMSE1** and **LabMSE1**) for the estimators with the default λ value. The effects of λ on our algorithms are investigated in Section 6.6. For the baselines, besides the usual versions (**LogME** and **TransRate**) that are computed from the extracted features and the target labels, we also consider the versions where they are computed from the dummy labels and the target labels (named **LabLogME** and **LabTransRate**). As in previous

work [Huang et al., 2022], we divide the target label values into equal-sized bins (five bins in our case) to compute TransRate and LabTransRate.

6.1 TRANSFER BETWEEN DIFFERENT DOMAINS

This experiment considers the general case where source models are trained on one dataset (OpenMonkey) and then transferred to another (CUB-200-2011). Specifically, we train a source model for each of the 17 keypoints of the OpenMonkey dataset and transfer them to each of the 15 keypoints of the CUB-200-2011 dataset, resulting in a total of 255 final models. Since each keypoint consists of x and y positions, all source and target tasks in this experiment have two dimensional labels. The actual MSEs of these models are computed on the respective test sets and then used to calculate the Pearson correlation coefficients with the transferability estimators. In this experiment, LabMSE0, LabMSE1, LabLogME, and LabTransRate are computed from the dummy source labels and the actual target labels.

Results for this experiment are in Table 1. In this setting, TransRate and LabTransRate perform poorly, while our methods are equal or better than LogME and LabLogME in all cases, especially when using $\lambda = 1$ (LinMSE1) or dummy labels (LabMSE0 and LabMSE1). The results show our approaches improve up to 25.9% in comparison with SotA (LogME) while being 9.4% better on average.

We also observe that adding a regularizer in LabMSE1 does not significantly change its result compared to LabMSE0. This is because the labels are already scaled to $[0, 1]$, so a regularizer with $\lambda = 1$ has negligible effect on the scores. We will observe the same effect in subsequent experiments (see Section 6.6 for experiments on the effects of λ). It is also surprising that LabMSE0 and LabMSE1 are better than LinMSE0 and LinMSE1. One possible explanation for this phenomenon is that the dummy labels (i.e., body parts of monkeys) give more information about the target labels (i.e., body parts of birds) than the extracted features.

In the supplementary, we also report additional results where both source and target tasks have 10-dimensional labels (i.e., each task predicts 5 keypoints simultaneously). We also achieve better correlations than the baselines in this case.

6.2 TRANSFER WITH SHARED-INPUTS TASKS

In this experiment, we consider the setting where the source and target tasks have the same inputs (the special setting in Section 4.2). Since images in our datasets contain multiple labels (15 keypoints for CUB-200-2011 and 17 keypoints for OpenMonkey), we can use any two different keypoints on the same dataset as source and target tasks. In total, we construct 210 source-target pairs for CUB-200-2011 and 272 pairs for OpenMonkey that all have the same source and target inputs but different labels. The labels for all tasks are also two dimensional real values.

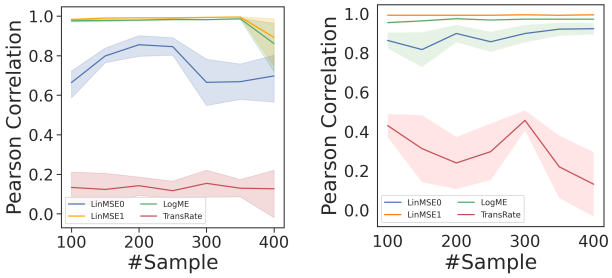


Figure 1: **Correlation coefficients with small target training set sizes** on CUB-200-2011 (left) and OpenMonkey (right). LinMSE1 and LogME are designed to avoid overfitting, but LinMSE1 is better than LogME in both datasets.

We repeat the experiment in Section 6.1 with these source-target pairs for CUB-200-2011 and OpenMonkey separately. The main difference in this experiment is that we use the *true* source labels (instead of dummy labels) when computing LabLogME, LabTransRate, LabMSE0, and LabMSE1. Under this setting, the LabMSE estimators here are the Shared Inputs Label MSE estimators in Definition 4.3. These estimators can be computed without any source models, and thus incurring very low computational costs in this setting.

Results for these experiments are in Table 2. In the results, both versions of TransRate perform poorly on CUB-200-2011, while TransRate is slightly better than LogME on OpenMonkey. In 5/6 rows of Table 2 (i.e., head re-training and half fine-tuning on CUB-200-2011, as well as all three settings for OpenMonkey), LabMSE0 and LabMSE1 both outperform LabLogME and LabTransRate, while LinMSE0 and LinMSE1 both outperform LogME and TransRate. In these five cases, the LinMSE methods achieve the best correlations overall. In the remaining setting where we transfer by full fine-tuning on the CUB-200-2011 dataset, all methods perform poorly. From these results, our approaches improve up to 94% in comparison with SotA (LogME) while being 26.8% better on average.

We also report in the supplementary additional results for each individual source task. The results show that our methods are consistently better than LogME, LabLogME, TransRate, and LabTransRate for most source tasks on both datasets. Furthermore, our methods are also better than these baselines when transferring to higher dimensional target tasks (tasks that predict 5 keypoints simultaneously and have 10-dimensional labels). These additional results further confirm the effectiveness of our approaches.

6.3 EVALUATIONS ON SMALL TARGET SETS

In many real-world transfer learning scenarios, the target set is usually small. This experiment will evaluate the effectiveness of the feature-based transferability estimators (LogME, TransRate, LinMSE0, and LinMSE1) in this small data regime where the number of samples is smaller than the feature dimension. For this experiment, we fix a source task

Table 2: **Correlation coefficients when transferring between tasks with shared inputs**. Bold numbers indicate best results in each row. Asterisks (*) indicate best results among the corresponding label-based or feature-based methods. Detailed correlation plots are in the supplementary. Our estimators improve up to 94% in comparison with SotA (LogME) while being 26.8% better on average.

Dataset	Transfer setting	Label-based method				Feature-based method			
		LabLogME	LabTransRate	LabMSE0	LabMSE1	LogME	TransRate	LinMSE0	LinMSE1
CUB-200-2011	Head re-training	0.547	0.008	0.916*	0.916*	0.889	0.029	0.921	**0.932***
	Half fine-tuning	0.400	0.006	0.536*	0.536*	0.560	0.006	**0.628***	0.607
	Full fine-tuning	**0.120***	0.001	0.056	0.056	0.099	0.100*	0.097	0.093
Open Monkey	Head re-training	0.890	0.666	0.973*	0.973*	0.646	0.711	0.949	**0.981***
	Half fine-tuning	0.615	0.340	0.754*	0.754*	0.391	0.488	**0.893***	0.787
	Full fine-tuning	0.569	0.269	0.705*	0.705*	0.352	0.439	**0.852***	0.739

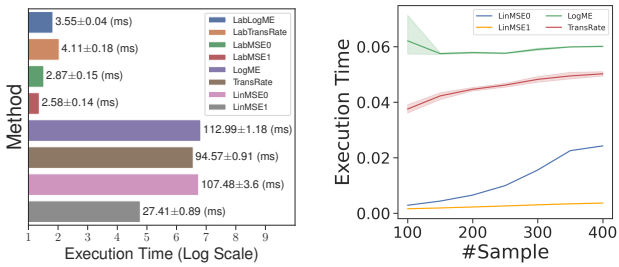


Figure 2: **Average running time** (in milliseconds) on CUB-200-2011 for the experiments in Section 6.2 (left) and Section 6.3 (right).

(*Belly* for CUB-200-2011 and *Right eye* for OpenMonkey) and transfer to all other tasks in the corresponding dataset using head re-training. For each target task, instead of using the full data, we randomly select a small subset of 100 to 400 images to perform transfer learning and to compute the transferability scores. The actual MSEs of the transferred models are still computed using the full target test sets.

Figure 1 compares the correlations of the 4 methods on different target set sizes between 100 and 400. The results are averaged over 10 runs with 10 different random seeds. From the figure, LogME and LinMSE1 are better than TransRate and LinMSE0. This is expected since LogME and LinMSE1 are designed to avoid overfitting on small data. Both LogME and LinMSE1 are also more stable, but LinMSE1 is slightly better than LogME on all dataset sizes.

6.4 EFFICIENCY OF OUR ESTIMATORS

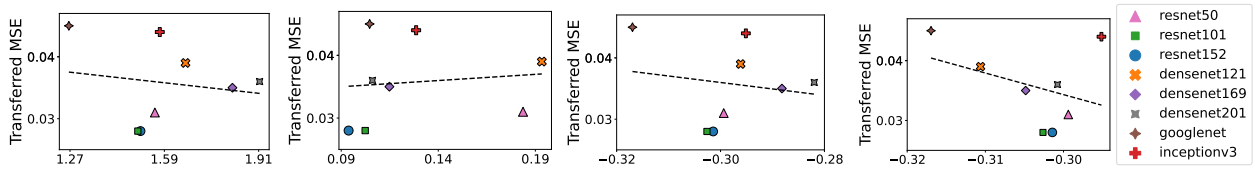
One of the main strengths of our methods is their efficiency due to the simplicity of training the Ridge regression head. In this experiment, we first use the settings in Section 6.2 to compare the running time of our methods with that of the baselines on the CUB-200-2011 dataset. Figure 2 (left) reports the results (averaged over 5 runs with different random seeds) for this experiment. From these results, our methods, LabMSE0, LabMSE1, LinMSE0, and LinMSE1, are all faster than the corresponding label-based or feature-based baselines. The figure also shows that LabMSE1 and LinMSE1 achieve the best running time among the label-based and feature-based methods respectively.

In Figure 2 (right), we also compare the average running time of the 4 transferability estimators using the CUB-200-2011 experiment in Section 6.3. This figure clearly shows that our methods, LinMSE0 and LinMSE1, are more computationally efficient than LogME and TransRate. Both results in Figure 2 show that LinMSE1 and LabMSE1 are significantly faster than other feature-based and label-based methods respectively. In these experiments, LinMSE1 and LabMSE1 converge faster than LinMSE0 and LabMSE0 respectively, and thus are more efficient.

6.5 SOURCE TASK SELECTION

Source task selection is important for applying transfer learning since the right source task can improve transfer learning performance [Nguyen et al., 2020]. In this experiment, we examine the application of our transferability estimation methods for selecting source tasks on the CUB-200-2011 dataset. We use the head re-training setting similar to Section 6.2, but fix one of the tasks as the target and choose the best source task from the rest of the task pool. We repeat this process for all 15 target tasks and measure the top-k matching rate of each transferability measure. The top-k matching rate is defined as $m_{\text{match}}/m_{\text{target}}$, where m_{target} is the total number of target tasks (15 in our case), and m_{match} is the number of times the selected source task gives a target model within the best k models. Here the best k models are determined by the actual test MSE on the target task.

From our experiment results, the *top*-5 *matching rates* obtained are: 0/15 (LabLogME), 0/15 (LabTransRate), **15/15** (LabMSE0), **15/15** (LabMSE1), **15/15** (LogME), 0/15 (TransRate), **15/15** (LinMSE0), and **15/15** (LinMSE1), while the *top*-3 *matching rates* are: 0/15 (LabLogME), 0/15 (LabTransRate), 0/15 (LabMSE0), 0/15 (LabMSE1), **15/15** (LogME), 0/15 (TransRate), **15/15** (LinMSE0), and **15/15** (LinMSE1). We also observe that all transferability scores perform equally poorly (0/15) on top-1 metric. These results show that our methods, LinMSE0 and LinMSE1, are comparable with LogME. When source models are not available (i.e., LogME, TransRate, LinMSE0, and LinMSE1 are not readily available), our methods (LabMSE0 and LabMSE1) are better than LabLogME when measured by the top-5 metric.



(a) LogME (b) TransRate (c) LinMSE0 (d) LinMSE1

Figure 3: **Test MSEs vs. transferability scores** when transferring from pre-trained classification models to a target regression task. A linear regression model (dashed line) is fitted to the points in each plot. Our methods give better fits than the baselines.

6.6 EFFECTS OF λ

In this experiment, we investigate the effects of λ on our proposed transferability estimators. We use the setting in Section 6.2 with the CUB-200-2011 dataset and vary the value of λ in [0, 100] for both LabMSE and LinMSE. Table 3 reports the results for all three transfer learning settings.

As noted in Section 6.1, adding a regularizer does not affect LabMSE much (unless λ is very large) since the labels used to compute LabMSE were already scaled to [0,1]. For head re-training, increasing λ improves the correlation for both LabMSE and LinMSE, with the best correlations at $\lambda = 100$. For half fine-tuning, increasing λ slightly improves the correlation for LabMSE, but decreases the correlation for LinMSE. In this setting, LabMSE is best at $\lambda = 100$, while LinMSE is best at $\lambda = 0$. For full fine-tuning, increasing λ slightly decreases the correlations for both LabMSE and LinMSE, with the best correlations achieved at $\lambda = 0$.

Notably, we can observe from Table 3 that varying λ does not significantly change the correlations. In the head re-training and half fine-tuning settings (where we have significant correlations for at least one transferability estimators), LabMSE *with any λ value* is better than LabLogME and LabTransRate, while LinMSE *with any λ value* is better than LogME and LabTransRate. These results show that our methods are generally robust to the value of λ and also better than the baselines for a wide range of λ values.

6.7 BEYOND REGRESSION

Although our paper mainly focuses on regression tasks, the main idea of using the negative regularized MSE of a Ridge regression model for transferability estimation goes beyond regression. In principle, this idea can be applied for transferring between classification tasks (in this case, we should train a linear classifier and use its regularized log-likelihood as the transferability estimator) or between a classification and a regression task.

In this section, we demonstrate that our idea can be applied for transferability estimation between a classification and a regression task. Particularly, we use 8 source models pre-trained on ImageNet [Deng et al., 2009] and transfer to a target regression task on the *dSprite* dataset [Matthey et al., 2017] using full fine-tuning. This setting is similar to You et al. [2021] where the target is a regression task with 4-dimensional labels: x and y positions, scale, and orientation.

Table 3: **Correlation coefficients for different values of** λ on CUB-200-2011. Bold numbers indicate best results in each column. Results of the baselines are given in the last 2 rows for comparison. When there are meaningful correlations (head re-training and half fine-tuning), our methods are better than the corresponding baselines for all λ values.

λ	Head re-training		Half fine-tuning		Full fine-tuning	
	LabMSE	LinMSE	LabMSE	LinMSE	LabMSE	LinMSE
0.0	0.9156	0.9205	0.5359	**0.6275**	**0.0558**	**0.0968**
0.5	0.9156	0.9316	0.5359	0.6069	**0.0558**	0.0921
1.0	0.9156	0.9317	0.5359	0.6071	**0.0558**	0.0929
2.0	0.9156	0.9321	0.5359	0.6079	**0.0558**	0.0941
5.0	0.9156	0.9331	0.5359	0.6092	**0.0558**	0.0956
7.0	0.9156	0.9336	0.5359	0.6095	0.0557	0.0959
10	0.9157	0.9342	0.5359	0.6096	0.0556	0.0961
20	0.9159	0.9358	0.5360	0.6092	0.0554	0.0959
50	0.9172	0.9380	0.5363	0.6072	0.0541	0.0948
75	0.9185	0.9391	0.5364	0.6060	0.0528	0.0943
100	**0.9198**	**0.9398**	**0.5365**	0.6051	0.0516	0.0938
(Lab)LogME	0.547	0.889	0.400	0.560	0.120	0.099
(Lab)TransRate	0.008	0.029	0.006	0.006	0.001	0.100

We compute the transferability scores from the extracted features and the labels of the target training set. More details about this experiment are in the supplementary.

From the results in Figure 3, the trends for LogME, LinMSE0, and LinMSE1 are correct (i.e., transferability scores have negative correlations with actual MSEs), while that of TransRate is incorrect. To compare these methods, we fit a linear regression to the points in each plot and compute its RMSE to these points, where we obtain: 6.12×10^{-3} (LogME), 6.16×10^{-3} (TransRate), 6.10×10^{-3} (LinMSE0), and $\mathbf{5.73} \times 10^{-3}$ (LinMSE1). These results show that LinMSE0 and LinMSE1 are better than LogME and TransRate.

7 CONCLUSION

We formulated transferability estimation for regression tasks and proposed the Linear MSE and Label MSE estimators, two simple but effective approaches for this problem. We proved novel theoretical results for these estimators, showing their relationship with the actual task transferability. Our extensive experiments demonstrated that the proposed approaches are superior to recent, relevant SotA methods in terms of efficiency and effectiveness. Our proposed ideas can also be extended to mixed cases where one of the tasks is a classification problem.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. In *European Conference on Computer Vision*, 2014.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019.

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *IEEE International Conference on Image Processing*, 2019.

Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. *Learning theory and kernel machines*, 2003.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, 2007.

Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. In *Advances in Neural Information Processing Systems*, 2021.

Xingyuan Bu, Junran Peng, Junjie Yan, Tieniu Tan, and Zhaoxiang Zhang. GAIA: A transfer learning system of object detection that fits your needs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Enyu Cai, Sriram Baireddy, Changye Yang, Melba Crawford, and Edward J Delp. Deep transfer learning for plant center localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

Aditya Deshpande, Alessandro Achille, Avinash Ravichandran, Hao Li, Luca Zancato, Charless Fowlkes, Rahul Bhotika, Stefano Soatto, and Pietro Perona. A linearized framework and a new benchmark for model selection for fine-tuning. *arXiv:2102.00084*, 2021.

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

Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3D human pose estimation: Motion to the rescue. In *Advances in Neural Information Processing Systems*, 2019.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, 2014.

Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Kshitij Dwivedi, Jiahui Huang, Radoslaw Martin Cichy, and Gemma Roig. Duality diagram similarity: A generic framework for initialization selection in task transfer learning. In *European Conference on Computer Vision*, 2020.

Ali Pourramezan Fard, Hojjat Abdollahi, and Mohammad Mahoor. ASMNet: A lightweight deep neural network for face alignment and pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Jiaji Huang, Qiang Qiu, and Kenneth Church. Exploiting a zoo of checkpoints for unseen tasks. In *Advances in Neural Information Processing Systems*, 2021.

Long-Kai Huang, Junzhou Huang, Yu Rong, Qiang Yang, and Ying Wei. Frustratingly easy transferability estimation. In *International Conference on Machine Learning*, 2022.

Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. Ranking neural checkpoints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Annual Conference on Learning Theory*, 2009.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing Sprites dataset, 2017. `https://github.com/deepmind/dsprites-dataset/`.

Cuong N Nguyen, Lam Si Tung Ho, Vu Dinh, Tal Hassner, and Cuong V Nguyen. Generalization bounds for deep transfer learning using majority predictor accuracy. In *International Symposium on Information Theory and Its Applications*, 2022.

Cuong V Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, 2020.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

Domenick D Poster, Shuowen Hu, Nathan J Short, Benjamin S Riggan, and Nasser M Nasrabadi. Visible-to-thermal transfer learning for facial landmark detection. *IEEE Access*, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 2021.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, 2019.

Max Schwarz, Hannes Schulz, and Sven Behnke. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In *IEEE International Conference on Robotics and Automation*, 2015.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2014.

Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, 2018.

Yang Tan, Yang Li, and Shao-Lun Huang. OTCE: A transferability metric for cross-domain cross-task representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Xinyi Tong, Xiangxiang Xu, Shao-Lun Huang, and Lizhong Zheng. A mathematical framework for quantifying transferability in multi-source transfer learning. In *Advances in Neural Information Processing Systems*, 2021.

Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. In *Advances in Neural Information Processing Systems*, 2020.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. *Technical Report*, 2011. `https://authors.library.caltech.edu/27452/`.

Boyu Wang, Jorge Mendez, Mingbo Cai, and Eric Eaton. Transfer learning via minimizing the performance gap between domains. In *Advances in Neural Information Processing Systems*, 2019.

Paul N Whatmough, Chuteng Zhou, Patrick Hansen, Shreyas Kolala Venkataramanaiah, Jae-sun Seo, and Matthew Mattina. FixyNN: Efficient hardware for mobile computer vision via transfer learning. In *Conference on Systems and Machine Learning*, 2019.

Yuan Yao, Abhiraj Abhiraj Mohan, Eliza Bliss-Moreau, Kristine Coleman, Sienna M Freeman, Christopher J Machado, Jessica Raper, Jan Zimmermann, Benjamin Y Hayden, and Hyun Soo Park. OpenMonkeyChallenge: Dataset and Benchmark Challenges for Pose Tracking of Non-human Primates. *bioRxiv*, 2021. `http://openmonkeychallenge.com/`.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, 2021.

Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.