KITE: A KERNEL-BASED IMPROVED TRANSFERABILITY ESTIMATION METHOD

Anonymous authors
Paper under double-blind review

ABSTRACT

Transferability estimation has emerged as an important problem in transfer learning. A transferability estimation method takes as inputs a set of pre-trained models and decides which pre-trained model can deliver the best transfer learning performance. Existing methods tackle this problem by analyzing the output of the pre-trained model or by comparing the pre-trained model with a probe model trained on the target dataset. However, neither is sufficient to provide reliable and efficient transferability estimations. In this paper, we present a novel perspective and introduce KITE, as a Kernel-based Improved Transferability Estimation method. KITE is based on the key observations that the separability of the pre-trained features and the similarity of the pre-trained features to random features are two important factors for estimating transferability. Inspired by kernel methods, KITE adopts *centered kernel alignment* as an effective way to assess feature separability and feature similarity. KITE is easy to interpret, fast to compute, and robust to the target dataset size. We evaluate the performance of KITE on a recently introduced large-scale model selection benchmark. The benchmark contains 8 source dataset, 6 target datasets and 4 architectures with a total of 32 pre-trained models. Extensive results show that KITE outperforms existing methods by a large margin for transferability estimation.

1 INTRODUCTION

Transfer learning has become the standard paradigm for addressing computer vision problems such as recognition (Sharif Razavian et al., 2014; Donahue et al., 2014), object detection (Girshick et al., 2014; Ren et al., 2015) and semantic segmentation (Long et al., 2015; Badrinarayanan et al., 2017). A generic representation can be transferred to a specific task by fine-tuning the pre-trained model (Hinton, 2007; Bengio, 2012). Compared with training from scratch, transfer learning leads to faster convergence (He et al., 2019; Raghu et al., 2019) and better performance (Chen et al., 2020). The effectiveness of transfer learning motivates researchers to train a large number of pre-trained models [1,2]. Practitioners can readily download these pre-trained models and specialize them for their own tasks. However, making transfer learning more accessible and reliable requires addressing one fundamental question:

Given a target dataset, which pre-trained model can deliver the best transfer learning performance?

There are several factors that make the problem of transferability estimation challenging. First, the number of pre-trained models can be huge. It is thus infeasible to fine-tune each pre-trained model. Second, the pre-trained models are trained with diverse architectures on different source datasets. Third, the target dataset can vary in characteristics and sizes. Although there have been major progresses in addressing these challenges, existing methods rely either on the output of the pre-trained model which is insufficient to provide accurate transferability estimations (Tran et al., 2019; Nguyen et al., 2020; You et al., 2021; Bao et al., 2019; Bolya et al., 2021) or on the similarity the pre-trained model with a probe model trained on the target dataset which is time-consuming (Dwivedi & Roig, 2019; Dwivedi et al., 2020).

[1] https://www.tensorflow.org/hub
[2] https://pytorch.org/hub/

In this paper, we present a novel perspective for transferability estimation by examining the usefulness of pre-trained features from two extremes. At one extreme, the pre-trained features are easy to separate. We find that this is the case if the target task is coarse-grained object classification and the separability of the pre-trained features is a strong indicator for transferability. At the other extreme, the pre-trained features behave similarly to random features. We find that this is case when the target task is fine-grained object classification and the dissimilarity of the pre-trained features to random features serves as a stronger hint. Together, our results show that although these two extremes are related; indeed, better separability generally indicates less similarity to random features. They still have different implications for transferability depending on the target tasks.

Based on the above analysis, we propose an improved transferability estimation method, called KITE, which examines both the separability of pre-trained features and the similarity of the pre-trained features to random features. In particular, KITE leverages *centered kernel alignment* (Cortes et al., 2012) which is a standard method for selecting kernels to measure feature separability and feature similarity. As a result, KITE provides an accurate assessment of transfer learning performance without training for a variety of target tasks. KITE achieves a new state-of-the-art on a large-scale model selection benchmark (Bolya et al., 2021) which contains a total of 32 pre-trained models. Qualitatively, KITE selects pre-trained models based on the source and target semantics. Quantitatively, KITE produces transferability estimation scores which are well correlated with the final fine-tuning accuracies as measured by Pearson correlation and Weighted Kendall's τ rank correlation (Vigna, 2015). In particular, KITE achieves an improvement of 11.90% over the state-of-the-art in terms of Pearson correlation.

Contributions. We highlight the following contributions:

- We address transferability estimation from a novel perspective by understanding the effects of feature separability and the similarity of the pre-trained features to random features. This perspective is advantageous since it introduces additional hints for transferability estimation without extra training cost. Our results also shed light on the impact of target tasks on transferability estimation.
- We propose KITE based on kernel methods for measuring feature separability and feature similarity. We find that feature separability is predictive of transferability only when the pre-trained features are *easy* to separate. When the object categories are hard to separate, the dissimilarity of the pre-trained features to random features serves as a better metric.
- We demonstrate that KITE outperforms existing transferability estimation methods on a large-scale model selection benchmark by a large margin in terms of correlation between the final transfer learning accuracies and the transferability estimation scores. KITE is also fast to compute and robust to target dataset variations.

2 RELATED WORK

Deep Transfer Learning. Transfer learning aims at leveraging the knowledge in a pre-trained model to boost the performance on a target dataset (Hinton, 2007; Bengio, 2012). In the standard way of conducting transfer learning, a deep neural network is first pre-trained on a large-scale source dataset such as ImageNet (Deng et al., 2009), then the pre-trained model is fine-tuned on the target dataset with discriminative learning. In particular, the pre-trained model can be trained in a supervised manner with the standard cross-entropy loss (Girshick et al., 2014; Ren et al., 2015; Kornblith et al., 2019b). Given unlabeled source datasets, the pre-trained models can also be trained without supervision with contrastive loss (Chen et al., 2020; He et al., 2020; Grill et al., 2020).

Transfer learning has been extensively studied recently due to its theoretical and practical importance. Several works aim at understanding the mechanism behind transfer learning by investigating questions including what is being transferred (Neyshabur et al., 2020), which layer is more transferable (Yosinski et al., 2014), what is the correlation between pre-training performance and transfer learning performance (Kornblith et al., 2019b) and the relation between loss functions and transfer learning performance (Kornblith et al., 2021). Other works focus on developing more sophisticated transfer learning methods with source target joint fine-tuning (Ge & Yu, 2017), instance-adaptive fine-tuning (Guo et al., 2019; 2020) or additional regularization terms (Xuhong et al., 2018).

Transferability Estimation. Given a target dataset, transferability estimation aims to select the most effective pre-trained model from a library of models for fine-tuning. H-score (Bao et al., 2019) proposes to estimate the transferability by seeking pre-trained features with high inter-class variance and low redundancy. NCE (Tran et al., 2019) and LEEP (Nguyen et al., 2020) estimate transferability by computing the likelihood the target dataset given the pre-trained model. Since likelihood is prone to over-fitting, LogME (You et al., 2021) proposes to leverage evidence maximization (Knuth et al., 2015) for transferability estimation. In particular, LogME computes the logarithm of maximum evidence of the target dataset based on the pre-trained model. More recently, a large-scale transferability estimation benchmark (Bolya et al., 2021) is proposed for evaluating different methods. PARC is also proposed in Bolya et al. (2021) to compute the Spearman correlation between the pre-trained features and target labels for transferability estimation. Methods for Taskonomy Model Selection (Zamir et al., 2018) can also be used for transferability estimation. RSA (Dwivedi & Roig, 2019) compares the features of the pre-trained model with a "probe" model which is trained on the target dataset. In particular, RSA computes the Pearson product-moment correlation coefficient between each pair of images. Spearman correlation is further used to compare the correlation coefficients produced by the pre-trained model and the "probe" model. DDS (Dwivedi et al., 2020) generalizes this idea by considering other metrics such as cosine distance and z-score.

3 BACKGROUND

Transferability Estimation. We follow the evaluation protocol in Bolya et al. (2021) for computing the transferability estimation score and comparing the performance of different transferability estimation methods. Assume that there are T target datasets and S source datasets. For each source dataset s, the corresponding pre-trained model is denoted as M_s. For each target dataset t, we sample n images which lead to a probe set P_n^t. Given a transferability estimation method $\mathcal{A}$, the transferability estimation score can be computed based on P_n^t as,

$$\alpha_{s,t} = \mathcal{A}(M_s, P_n^t) \tag{1}$$

Intuitively, $\alpha_{s,t}$ indicates the transferability of the pre-trained model M_s on the target dataset t. The ground-truth fine-tuning accuracy of the model M_s on the target dataset t is denoted as $w_{s,t}$. The performance of the transferability estimation method $\mathcal{A}$ is measured via Pearson correlation (Freedman et al., 2007) between $\alpha_{s,t}$ and $w_{s,t}$ for each pre-trained model,

$$TE(\mathcal{A}) = \frac{1}{T}\sum_{t=1}^{T} \text{Pearson Correlation}([\alpha_{s,t}], [w_{s,t}]) \tag{2}$$

where $[\alpha_{s,t}]$ and $[w_{s,t}]$ are lists of transferability estimation scores and ground-truth fine-tuning accuracies obtained by each pre-trained model s, respectively. A larger $TE(\mathcal{A})$ indicates that the transferability estimation score is well correlated with the transfer accuracy, i.e., better transferability estimation. Other metrics such as weighted version of Kendall's τ (Vigna, 2015) are also considered in the literature (You et al., 2021).

Kernel Theory. KITE is rooted in the standard theory of kernels in machine learning (Smola & Schölkopf, 1998). Consider a Hilbert space $\mathcal{F}$ which consists of functions from $\mathcal{X}$ to $\mathbb{R}$. $\mathcal{F}$ is a reproducing Kernel Hilbert Space (RKHS) if for each $\mathbf{x} \in \mathcal{X}$, the Dirac evaluation functional $\sigma_{\mathbf{x}} : \mathcal{F} \to \mathbb{R}$ is a bounded linear functional. For each RKHS, there exists a unique positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' \in \mathcal{X}$, there exist corresponding element $\phi(\mathbf{x}) \in \mathcal{F}$ and $\phi(\mathbf{x}') \in \mathcal{F}$ such that $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{F}} = k(\mathbf{x}, \mathbf{x}')$. With kernels, instead of computing inner products in the high-dimensional feature space $\mathcal{F}$, we can operate in the input space $\mathcal{X}$. There are several kernel functions such as linear kernel, Gaussian kernel, polynomial kernel and Laplacian kernel (Smola & Schölkopf, 1998). In particular, the Gaussian kernel, also called the *Radial Basis Kernel*, is defined as,

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x} - \mathbf{y}\|}{2\sigma^2}) \tag{3}$$

where σ is a free parameter.

Centered Kernel Alignment. Given a set of samples $S = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, the kernel matrix $\mathbf{K}$ can be computed by applying the kernel to each pair of samples, i.e., $\mathbf{K}[i, j] = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Different kernel functions give different similarity measures. Kernel alignment was introduced in Cristianini et al. (2001) as a means of comparing two kernels. Given two kernels $k_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we first compute the kernel matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ on the samples $S = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, the alignment of k_1 and k_2 is defined as,

Definition 3.1 (Alignment (Cristianini et al., 2001))**.**

$$\rho(\mathbf{K}_1, \mathbf{K}_2) = \frac{\langle \mathbf{K}_1, \mathbf{K}_2 \rangle_F}{\sqrt{\langle \mathbf{K}_1, \mathbf{K}_1 \rangle_F \langle \mathbf{K}_2, \mathbf{K}_2 \rangle_F}} \tag{4}$$

where $\langle\rangle_F$ is the Frobenius inner product.

Kernel alignment has been applied for selecting kernels or combining multiple kernels (Cristianini et al., 2001). However, kernel alignment does not correlate well the classification performance (Cortes et al., 2012). This limitation is addressed via centered kernel alignment (Cortes et al., 2012) which normalizes the features in the feature space. In particular, the feature mapping is centered by subtracting from the mean as $\phi(\mathbf{x}) - \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i)$. The corresponding centered kernel matrix $\mathbf{K}^c$ can be computed from the uncentered kernel matrix $\mathbf{K}$,

Definition 3.2 (Centered Kernel Matrix)**.**

$$\mathbf{K}^c = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right] \mathbf{K} \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right] \tag{5}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{1} \in \mathbb{R}^{n \times 1}$ is a matrix of ones.

With the definition of centered kernel matrix, Centered Kernel Alignment (CKA) is defined as follows,

Definition 3.3 (Centered Kernel Alignment (CKA) (Cortes et al., 2012))**.**

$$\text{CKA}(\mathbf{K}_1^c, \mathbf{K}_2^c) = \frac{\langle \mathbf{K}_1^c, \mathbf{K}_2^c \rangle_F}{\sqrt{\langle \mathbf{K}_1^c, \mathbf{K}_1^c \rangle_F \langle \mathbf{K}_2^c, \mathbf{K}_2^c \rangle_F}} \tag{6}$$

CKA has shown to have a better theoretical guarantee and is better correlated with performance of the kernel on downstream tasks (Cortes et al., 2012).

4 KITE

We begin by discussing the motivation, and then introduce KITE as a kernel-based improved transferability estimation method. Next, we give the interpretation and complexity of KITE. Finally, we compare KITE with existing transferability estimation methods.

Motivation. The usefulness of the pre-trained models naturally depends on the characteristics of the target dataset (Neyshabur et al., 2020; Kornblith et al., 2019b). For clarity, the pre-trained features are used to refer to the features obtained on the target dataset with the pre-trained model. At one extreme, the pre-trained features are useful and already capture the inter-class variations. At the other extreme, the pre-trained features are similar to random features which fail to capture inter-class separability. Depending on the target task, the usefulness of the pre-trained features can naturally locate anywhere between the two extremes. We find that if the target task is coarse-grained object classification, the pre-trained features are generally easy to separate and the separability is a strong indicator for transferability. However, if the target task is fine-grained object classification, the pre-trained features have poor separability and the dissimilarity of the pre-trained features to random features is a more effective metric for estimating transferability (see Section 5.3).

The above observations lead to the following criteria for estimating transferability which takes the two extreme cases into consideration. A pre-trained model is preferred if **1)** it behaves differently from a *random* network and **2)** it captures inter-class variations. As we will show, better separability indeed means less similarity to random features. However, depending on the target task, the two criteria are still complementary to each other and can be naturally combined for more accurate transferability estimations. The proposed KITE leverages CKA for computing feature separability and feature similarity which can be applied to different target datasets.

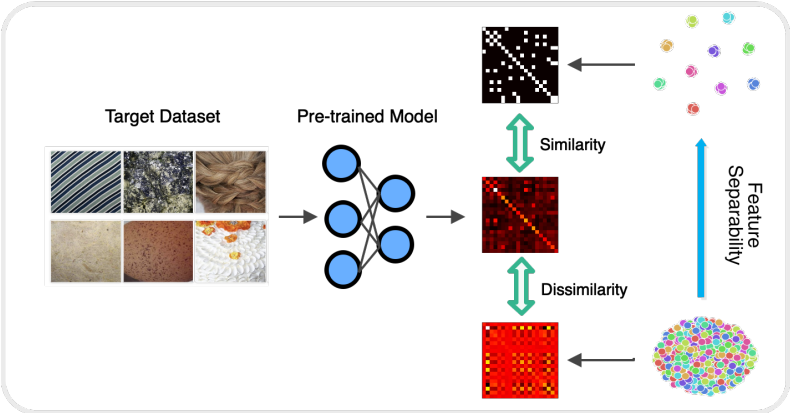


Figure 1: **KITE considers the separability of pre-trained features and the dissimilarity of the pre-trained features to random features for transferability estimations.** The pre-trained model is first used to generate features for the target dataset. Then, we compute the pre-trained feature kernel matrix, the random feature kernel matrix and the target kernel matrix. CKA is used to compare the (dis)similarity of the pre-trained feature kernel matrix to the random feature kernel matrix and the target kernel matrix.

Computing KITE for Transferability Estimation. The main idea of KITE is to compute the separability of the pre-trained features and the similarity of the pre-trained features to random features based on CKA. KITE measures how *close* the pre-trained features to the ideal features which perfectly separate the target categories and how *far* the pre-trained features from random features. The random features are generated by using an untrained network on the target dataset. Please see Figure 1 for an overview of KITE.

Given a probe set $P_n^t = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)..., (\mathbf{x}_n, y_n)\}$ sampled from the target dataset t, a model M_s pre-trained on the source dataset s and an untrained random network M_{random}. M_s and M_{random} are of the same architecture. Depending on different network initializations, the parameters of M_{random} are different. For simplicity, we assume M_{random} is given which is initialized with some initialization method and the effect of initializations will be investigated in Section 5.2. We first generate the feature vectors of the probe set using M_s and M_{random} as $F_s = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_n\}$ and $F_{random} = \{\tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2, ..., \tilde{\mathbf{f}}_n\}$, respectively. The feature vector is the output of the model before the classification layer. A kernel function, such as linear kernel or Gaussian kernel, is applied on the features to generate the pre-trained feature kernel matrix $\mathbf{K}_s \in \mathbb{R}^{n \times n}$ and the random feature kernel matrix $\mathbf{K}_{random} \in \mathbb{R}^{n \times n}$. Then the label y_i is converted into one-hot representation. We use $\mathbf{y}_i \in \mathbb{R}^{K \times 1}$ to denote the one-hot encoding of y_i, where K is the number of classes. We compute the target kernel matrix $\mathbf{K}_Y$ as $\mathbf{K}_Y[i, j] = \mathbf{y}_i^T \mathbf{y}_j$. Thus, $\mathbf{K}_Y[i, j]$ is 1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class, otherwise $\mathbf{K}_Y[i, j] = 0$. Intuitively, $\mathbf{K}_Y$ is the ideal kernel matrix which captures the ground-truth inter-class variations. With the pre-trained feature kernel matrix, the random feature kernel matrix and the target kernel matrix, KITE can be defined as follows,

Definition 4.1 (KITE)**.** *Given a probe set P_n^t of size n sampled from the target dataset t, a pre-trained model M_s, an untrained random network M_{random},* KITE *is defined as,*

$$\text{KITE}(M_s, M_{random}, P_n^t) = \frac{\text{CKA}(\mathbf{K}_s, \mathbf{K}_Y)}{\text{CKA}(\mathbf{K}_s, \mathbf{K}_{random})} \tag{7}$$

We refer to $\text{CKA}(\mathbf{K}_s, \mathbf{K}_Y)$ as *Target Alignment* (TA) which captures feature separability and $\text{CKA}(\mathbf{K}_s, \mathbf{K}_{random})$ as *Random Alignment* (RA) which indicates the difference between the pre-trained features and random features. A high KITE score implies that the pre-trained features are different from random features or the pre-trained features are easy to separate. While both Target Alignment and Random Alignment are indicative of transferability, we show that neither of them can decide the transferability of the pre-trained model alone. KITE considers both these two factors which can provide more accurate transferability estimations.

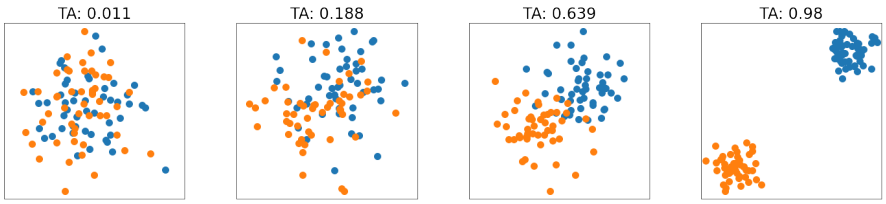
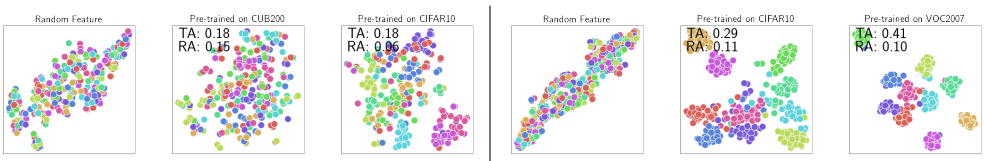


Figure 2: **TA captures separability of the features**. We validate the effectiveness of TA by generating multiple synthetic datasets. The datasets are generated by sampling from a mixture of two Gaussian distributions with different means. Clearly, the TA score correlates well with feature separability.



(a) Target: Stanford Dogs (b) Target: Caltech-101

Figure 3: **TA and RA uncover different patterns in the feature space.** TA can detect feature separability while RA can expose sample-wise similarity.

Interpretation. Figure 2 shows that TA scores correlate well with the separability the features. We create different datasets by sampling from a mixture of two Gaussian distributions with different means and unit variance. By changing the means of the distributions, we can create datasets with different degrees of separability. The TA score increases as the features become more separable. Figure 3 shows the t-SNE (Van der Maaten & Hinton, 2008) visualizations of the features extracted by different pre-trained models on Stanford Dogs and Caltech-101. It is worth noting that Stanford Dogs is a fine-grained classification task while Caltech-101 is a coarse-grained classification task. It can be observed that the features of Caltech-101 are easier to separate and achieve a higher TA score. This is a general phenomenon for coarse-grained classification tasks as shown in Section A.2 of the Appendix. The behavior of RA is more interesting. On Stanford Dogs, although the features produced by the models pre-trained on CUB200 and CIFAR10 are both hard to separate, the features produced by the model pre-trained on CIFAR10 are far from random: they nearly uncover similarities between the samples from the same class. This is captured via the RA score. In this case, the TA score is not informative since it imposes a much stronger requirement on the feature space. Although TA and RA are generally negatively correlated (see Section A.2 of the Appendix), a small RA score does not necessarily mean a large TA score (see Figure 3 (a)). We also find that it is generally infeasible to compare the scores across datasets due to the differences of the datasets. KITE takes both TA and RA into consideration which can deal with target datasets with different characteristics. Section 5.3 further shows that TA is particularly effective if the target task is coarse-grained object classification and RA is effective when the target task is fine-grained classification.

Complexity. KITE only requires two forward passes to compute the features of the pre-trained model and the random model, which is inevitable for most of the transferability estimation methods. Given n samples with feature dimension d, the complexity of computing the kernel matrices is $\mathcal{O}(n^2 d)$. Given that we usually sample a small probe set and the feature dimension is in orders of hundreds or thousands, KITE incurs negligible computational cost.

Connection to Existing Methods. The existing transferability estimation methods roughly fall into two broad categories: **1)** the estimation is only based on the output the pre-trained model (Tran et al., 2019; Nguyen et al., 2020; You et al., 2021; Bao et al., 2019; Bolya et al., 2021) or **2)** the estimation is based on the comparison of the pre-trained model with a probe model trained on the target dataset (Dwivedi & Roig, 2019; Dwivedi et al., 2020). Different from existing methods, KITE takes a novel perspective by considering both feature separability and the similarity of the pre-trained features to random features. As we will show, KITE provides much more accurate transferability estimations while being efficient to compute.

5 EXPERIMENTS

Benchmark. We adopt the transferability estimation benchmark proposed in Bolya et al. (2021) for evaluation. The transferability estimation benchmark consists of 8 source datasets, 6 target datasets and 4 architectures. The source datasets include ImageNet 1k (Deng et al., 2009), VOC2007 (Everingham et al., 2010), Caltech101 (Fei-Fei et al., 2006), CIFAR10 (Krizhevsky et al., 2009), NA Birds (Van Horn et al., 2015), CUB200 (Wah et al., 2011), Oxford Pets (Parkhi et al., 2012) and Stanford Dogs (Khosla et al., 2011). The target datasets include NA Birds, Stanford Dogs, Caltech101, CIFAR10, Oxford Pets, CUB200. The architectures include ResNet-50 (He et al., 2016), ResNet-18 (He et al., 2016), GoogLeNet (Szegedy et al., 2015) and AlexNet (Krizhevsky et al., 2012). There are a total of 32 pre-trained with all the combinations of source datasets and architectures. Please refer to Bolya et al. (2021) for more details. For each target dataset, we aim to select the most effective pre-trained model from all the pre-trained models for fine-tuning.

Baselines. We consider the following baselines which include the state-of-the-art methods for transferability estimation: *Probability-based Methods*: NCE (Tran et al., 2019), LEEP (Nguyen et al., 2020) and LogME (You et al., 2021). *Feature-based Methods*: RSA (Dwivedi & Roig, 2019), DDS (Dwivedi et al., 2020), H-Score (Bao et al., 2019) and PRAC (Bolya et al., 2021). *Heuristic-based Methods*: Logistic (Bolya et al., 2021), 1-NN CV (Bolya et al., 2021), 5-NN CV (Bolya et al., 2021), and Heuristic (Bolya et al., 2021). In particular, Logistic trains a logistic classifier on 50% the probe set and uses the accuracy on the other half as the score. K-NN CV (K = 1 or 5) adopts K-nearest neighbors with leave-one out cross-validation. Heuristic simply considers the number of layers in the pre-trained model ℓ_s, the size of the training dataset $|D_s|$ and target dataset $|D_t|$,

$$\text{Heuristic} = \ell_s + \log(|D_s| + |D_t|) \tag{8}$$

Metrics. We consider Pearson Correlation which takes all the pre-trained models into account for comparing the performance as in Bolya et al. (2021). We also consider weighted version of Kendall's τ (Vigna, 2015) which focuses more on the top performing models as in You et al. (2021).

Implementation Details. We follow the implementation from Bolya et al. (2021) for a fair comparison. The input images are resized to 224 $\times$ 224 for all the datasets. The probe set is constructed in a way that there are at least 2 examples for each class. The size of the probe set is 500. The feature dimension is reduced to 32 using principal component analysis (PCA) (Abdi & Williams, 2010). The pre-trained models and fine-tuning accuracies are provided by Bolya et al. (2021). For NCE, LEEP, RSA, DDS, Logistics, 1-NN CV, 5-NN CV and Heuristics, we use the implementations from Bolya et al. (2021). For LogME, we use the implementation provided by the original authors [3]. We implement KITE based on the framework provided by Bolya et al. (2021) in Python. No other heuristics are added to the methods for a fair comparison. We use linear kernel in the experiments as we will show that KITE is robust to the choices of kernels. The experiments are repeated for 3 runs with different random seeds. The sampled probe set and the initialization of the random network are different across runs. We report both average performance and standard deviation. All the experiments are done on one NVIDIA GeForce GTX GPU. The code is attached in the Supplementary and will be made public.

5.1 THE RESULTS OF TRANSFERABILITY ESTIMATION

Table 1 shows the comparison of KITE with all the baselines. Both in terms of Pearson Correlation (**Mean PC**) and weighted Kendall's τ (**Mean** τ), the proposed KITE improves the state-of-the-art by a large margin. In particular, KITE improves the **Mean PC** by 11.90% over 1-NN CV and **Mean** τ by 3.38% over 1-NN CV. This shows that KITE can accurately estimate the transferability of the pre-trained model. In terms of computational time, KITE is comparable to other competitive baselines while providing a significant better transferability estimation performance. It is worth noting that 1-NN CV and 5-NN CV does not scale well in terms of the number of samples since they rely on k-nearest neighbors algorithm. In Section 5.2, we investigate the performance of using Target Alignment and Random Alignment separately. Section 5.3 shows the effect of target datasets on the performance of KITE. Section 5.2 further shows that KITE is robust to the size of the probe set and feature dimension.

[3] https://github.com/thuml/LogME

Method	Need Training	Input	Time (ms) ↓	Mean PC (%) ↑	Mean τ ↑
NCE	No	$P_s(\mathbf{x}), y$	12.5	2.21 ± 0.52	0.19 ± 0.00
LEEP	No	$P_s(\mathbf{x}), y$	7.8	10.83 ± 0.13	0.20 ± 0.00
LogME	No	$M_s(\mathbf{x}), y$	2139.4	55.30 ± 0.41	0.47 ± 0.00
H-Score	No	$M_s(\mathbf{x}), y$	83.1	55.66 ± 0.54	0.55 ± 0.00
RSA	Yes	$M_s(\mathbf{x})$	222.4	5.37 ± 0.57	0.03 ± 0.01
DDS	Yes	$M_s(\mathbf{x})$	188.8	10.58 ± 0.32	0.03 ± 0.00
PARC	No	$M_s(\mathbf{x}), y$	149.1	59.15 ± 1.17	0.54 ± 0.04
Logistic	Yes	$M_s(\mathbf{x}), y$	338.0	58.31 ± 2.39	0.57 ± 0.01
1-NN CV	Yes	$M_s(\mathbf{x}), y$	123.8	60.68 ± 1.84	0.59 ± 0.02
5-NN CV	Yes	$M_s(\mathbf{x}), y$	138.2	59.73 ± 1.75	0.58 ± 0.01
Heuristic	No	N/A	N/A	50.76 ± 0.00	0.53 ± 0.00
KITE	No	$M_s(\mathbf{x}), M_{random}(\mathbf{x}), y$	135.7	**67.90 ± 0.70**	**0.61 ± 0.02**

Table 1: **KITE improves upon the existing methods for transferability estimation by a large margin both in terms of Pearson Correlation (Mean PC) and weighted Kendall's τ (Mean Kendall's τ).** KITE is also *fast* to compute and requires no training.

Method	Need Training	Input	Time (ms) ↓	Mean PC (%) ↑	Mean τ ↑
RA	No	$M_s(\mathbf{x}), M_{random}(\mathbf{x})$	83.9	56.33 ± 0.33	0.53 ± 0.01
TA	No	$M_s(\mathbf{x}), y$	43.5	14.87 ± 0.21	0.19 ± 0.01
HSIC	No	$M_s(\mathbf{x}), M_{random}(\mathbf{x})$	90.9	-4.04 ± 1.29	0.12 ± 0.01
KITE	No	$M_s(\mathbf{x}), M_{random}(\mathbf{x}), y$	135.7	**67.90 ± 0.70**	**0.61 ± 0.02**

Table 2: **KITE improves upon Random Alignment (RA), Target Alignment (TA) and HSIC by a large margin both in terms of Pearson correlation (Mean PC) and weighted Kendall's τ (Mean Kendall's τ).**

5.2 ABLATION STUDIES

KITE vs. other alternatives. Naturally, two alternatives are TA and RA. We also consider a third alternative called *Hilbert-Schmidt Independence Criterion* (HSIC) which was proposed in Gretton et al. (2005) as a measure of dependence between two random variables X and Y. Assume that $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and $\{\mathbf{y}_1, ..., \mathbf{y}_n\}$ are drawn from the joint distribution (X, Y). $\mathbf{K}$ is the kernel matrix computed on $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and $\mathbf{L}$ is the kernel matrix computed on $\{\mathbf{y}_1, ..., \mathbf{y}_n\}$. The empirical HSIC can be computed as $\text{HSIC}(X, Y) = \text{HSIC}(\mathbf{K}, \mathbf{L}) = \frac{1}{(n-1)^2}\,\text{Tr}(\mathbf{KHLH})$, where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix. One idea is to use $\text{HSIC}(\mathbf{K}_s, \mathbf{K}_{random})$ for transferability estimation. Table 2 shows that KITE outperforms all the alternatives by a large margin. There are two main reasons: **1)** Different from RA and HSIC, KITE considers target labels to compute feature separability. **2)** Different from TA, KITE considers the differences of the pre-trained features with random features which can examine if the pre-trained features capture sample-wise similarity. Noted that HSIC performs badly since it does not normalize the kernel matrices as in CKA (Kornblith et al., 2019a).

What is the effect of initialization of the random network? We consider three commonly used initializations: Xavier normal (Glorot & Bengio, 2010), He normal (He et al., 2015) and He uniform (He et al., 2015). Table 4 shows that initializations of the untrained network have little impact on the performance of KITE. Intuitively, initializations have more influence on the learning of the model and no initializations implicitly capture data similarity.

Does the size of the probe set matter? For each target dataset, we vary the probe set size in {100, 500, 1000, 2000}. Figure 4 a) shows that KITE achieves the best results across all the cases. Generally, we observe that across all the methods the performance improves as the probe set size increases. However, the performance does not increase much as the probe set grows even larger. This indicates that more target samples may introduce noises which is challenging for all transferability estimation methods. How to leverage large probe sets for transferability estimation is an interesting future direction.

Does the feature dimension matter? We use principal component analysis (PCA) (Abdi & Williams, 2010) to reduce the feature dimension to 32, 64 and 128. We also consider using the original feature dimension which is denoted as `full`. Figure 4 (b) shows that KITE is robust to the change of feature dimension and achieves the best results across all the cases.

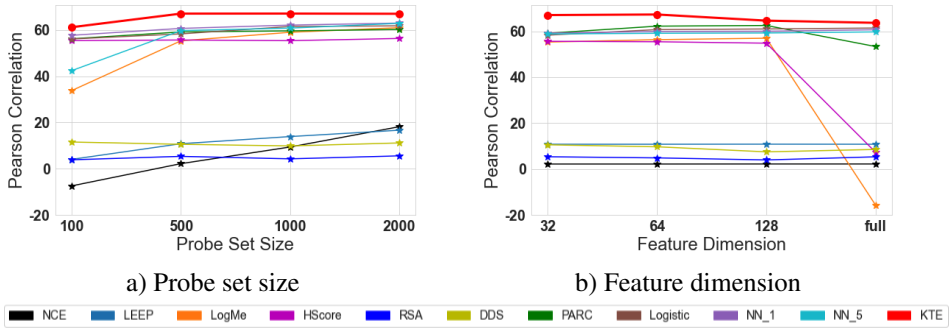


Figure 4: **KITE is robust to the probe set size and feature dimension.** a) The change of Pearson correlation as we change the size of the probe set. b) The change of Pearson correlation as we change the feature dimension.

Kernel	Mean PC (%) ↑	Mean τ ↑
Linear	67.90 ± 0.70	0.61 ± 0.02
Laplacian	65.51 ± 0.62	0.61 ± 0.01
Gaussian	66.66 ± 0.47	0.62 ± 0.01

Table 3: **KITE is robust to the choices of kernels.** We consider linear, Laplacian and Gaussian kernel for computing the kernel matrices.

Init	Mean PC (%) ↑	Mean τ ↑
He Normal	67.90 ± 0.70	0.61 ± 0.02
He Uniform	66.37 ± 1.08	0.63 ± 0.01
Xavier Normal	68.42 ± 0.56	0.62 ± 0.01

Table 4: **KITE is robust to different initializations of the random network**. We consider Xavier normal, He normal and He uniform.

Method	Fine-Grained Target Datasets	Coarse-Grained Target Datasets
RA	79.83 ± 0.78	32.83 ± 1.43
TA	-6.02 ± 0.52	50.04 ± 0.50
KITE	79.10 ± 0.89	56.71 ± 0.51

Table 5: **TA is effective when the target task is coarse-grained classification and RA is effective when the target task is fine-grained classification. KITE leverages the advantages of both metrics to provide more accurate transferability estimation.** The Pearson correlations achieved by different methods are shown.

What are the impacts of different kernel functions? We investigate the choices the kernel functions on the performance of KITE. We consider linear, Gaussian and Laplacian kernel. Table 3 shows that KITE is robust to the choices of kernel functions. This allows KITE to leverage simple linear kernels to evaluate the kernel matrices which is efficient to compute.

5.3 THE EFFECT OF TARGET DATASETS

The effects of target datasets are largely overlooked in the literature of transferability estimation. We find that TA and RA behave rather differently on fine-grained and coarse-grained classification tasks. We use TA to measure and rank the separability of the features. The most separable datasets are three coarse-grained classification tasks: CIFAR10, Oxford Pets and Caltech-101. The least separable datasets are three fine-grained classification tasks: CUB-200, Stanford Dogs and NABirds. Detailed scores are shown in Section A.2 of the Appendix. Table 5 shows the average Pearson correlations achieved by TA, RA and KITE. First, it can be observed that TA is particularly effective for coarse-grained classification tasks. Second, when the features are hard to separate as in the case of fine-grained classification, RA emerges to be more effective. KITE combines TA and RA which can provide accurate transferability estimations regardless of the characteristics of the target tasks. More discussions can be found in Section A.2 of the Appendix.

6 CONCLUSION

We bring a new perspective and propose an effective method called KITE for transferability estimation. KITE estimates transferability by assessing feature separability and comparing the pre-trained model with a random network based on centered kernel alignment. KITE is fast to compute, easy to interpret and robust to the target dataset variations. Extensive experiments demonstrate the effectiveness of KITE over the existing transferability estimation methods.

REFERENCES

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

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 *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2309–2313. IEEE, 2019.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13:795–828, 2012.

Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz Kandola. On kernel-target alignment. *Advances in neural information processing systems*, 14, 2001.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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*, pp. 647–655. PMLR, 2014.

Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12387–12396, 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*, pp. 497–513. Springer, 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1086–1095, 2017.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pp. 63–77. Springer, 2005.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4805–4814, 2019.

Yunhui Guo, Yandong Li, Liqiang Wang, and Tajana Rosing. Adafilter: Adaptive filter fine-tuning for deep transfer learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4060–4066, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4927, 2019.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Geoffrey E Hinton. To recognize shapes, first learn to generate images. *Progress in brain research*, 165:535–547, 2007.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, volume 2. Citeseer, 2011.

Kevin H Knuth, Michael Habeck, Nabin K Malakar, Asim M Mubeen, and Ben Placek. Bayesian evidence and model selection. *Digital Signal Processing*, 47:50–67, 2015.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019a.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019b.

Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? *Advances in Neural Information Processing Systems*, 34, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pp. 7294–7305. PMLR, 2020.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

Alex J Smola and Bernhard Schölkopf. *Learning with kernels*, volume 4. Citeseer, 1998.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1395–1405, 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 595–604, 2015.

Sebastiano Vigna. A weighted correlation index for rankings with ties. In *Proceedings of the 24th international conference on World Wide Web*, pp. 1166–1176, 2015.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pp. 2825–2834. PMLR, 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

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*, pp. 12133–12143. PMLR, 2021.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

A APPENDIX

We propose both a novel perspective and an effective method for transferability estimation. Our core idea has two folds: **1)** computing the separability of pre-trained features and **2)** assessing the dissimilarity of the pre-trained features to random features. We demonstrate the state-of-the-art performance for transferability estimation on a large-scale benchmark. In this appendix, we include more details and results:

- We show the models selected by KITE for each target dataset in Section A.1.
- We show the TA and RA scores for each target dataset and have a detailed analysis in Section A.2.
- We show the results of the methods on each target dataset separately in Section A.3.
- We discuss the limitations and extensions of KITE in Section A.4.

A.1 WHICH PRE-TRAINED MODEL IS SELECTED BY KITE?

In Figure 5, we show the selection of KITE for each target dataset. It can be observed that KITE selects the pre-trained model based on the semantics of the source dataset and the target dataset. The selections are also well matched to human intuition. This also indicates that transferability estimation is particularly useful when the source of the pre-trained model is unknown or the semantic similarity between the source and the target is hard to quantify.

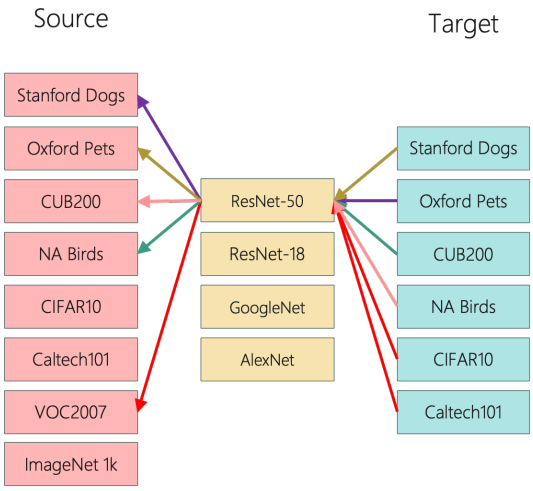


Figure 5: **KITE selects the pre-trainded model based on the semantics of the source dataset and the target dataset.**

A.2 TARGET ALIGNMENT VS. RANDOM ALIGNMENT

Figure 6 shows the TA and RA scores of the features extracted by a ResNet-18 pre-trained on different source datasets on the target datasets. There are several interesting observations from the results. The first observation is the TA scores are higher for coarse-grained classification tasks (Caltech-101, CIFAR10 and Oxford Pets) compared with fine-grained classification tasks (NA Birds, CUB-200 and Stanford Dogs). *We use the highest score each target dataset can achieve for ranking the separability*. This indicates that the features of coarse-grained classification tasks are inherently easier to separate. The second observation is that the TA scores and RA scores are negatively correlated. Intuitively, this means better separability indeed indicates less similarity to random features. Still, TA and RA have different implications on the properties of the feature space and different effects for transferability estimation based on the target task.

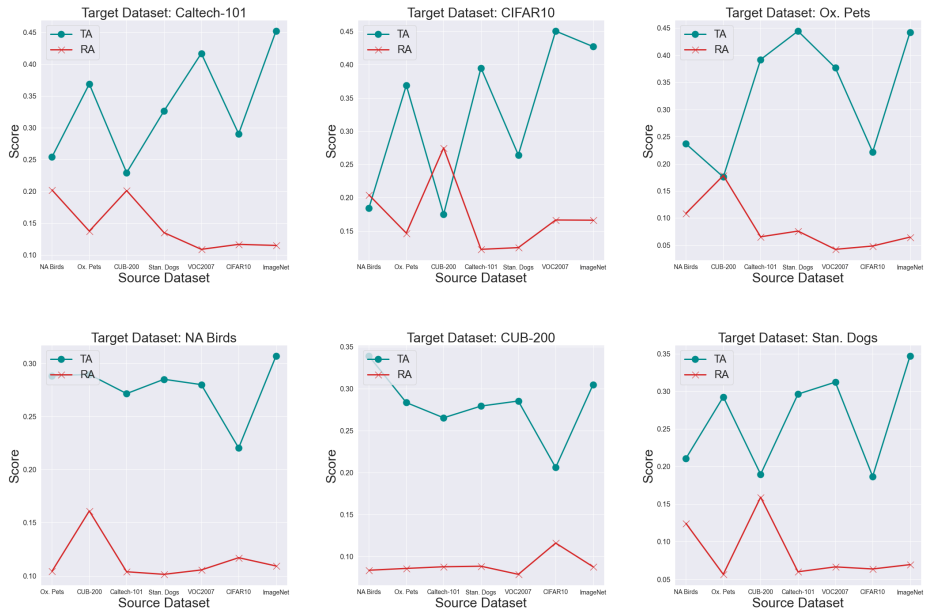


Figure 6: The TA scores for coarse-grained datasets are higher than fine-grained datasets. Also, TA and RA have a negative correlation.

A.3 Results for Each Target Dataset

Table 6 summarizes the results of the methods on each dataset.

Method	Stan. Dogs	Ox. Pets	CUB 200	NA Birds	CIFAR 10	Caltech 101	**Mean PC (%)** ↑
NCE	-2.46 ± 2.61	36.16 ± 4.57	9.97 ± 2.31	24.66 ± 11.48	75.20 ± 4.32	-14.37 ± 1.24	2.21 ± 0.52
LEEP	30.68 ± 0.34	39.37 ± 1.98	-1.52 ± 1.14	-17.37 ± 0.94	76.91 ± 1.62	-21.19 ± 0.50	10.83 ± 0.13
LogME	81.12 ± 0.51	72.09 ± 1.97	42.75 ± 0.31	47.52 ± 0.12	47.40 ± 2.57	65.58 ± 1.15	55.30 ± 0.41
H-Score	65.98 ± 2.43	71.61 ± 1.89	75.84 ± 3.83	65.62 ± 1.33	46.45 ± 2.39	64.23 ± 0.95	55.66 ± 0.54
RSA	-46.58 ± 0.83	-1.82 ± 1.73	-42.15 ± 2.23	-58.77 ± 1.82	19.35 ± 2.88	-23.00 ± 0.44	5.37 ± 0.57
DDS	-38.83 ± 0.72	11.96 ± 2.33	-40.14 ± 1.94	-59.29 ± 2.12	22.83 ± 2.11	-17.83 ± 0.79	10.58 ± 0.32
PARC	67.11 ± 1.94	47.29 ± 18.43	73.44 ± 3.49	74.83 ± 1.07	44.31 ± 4.20	59.22 ± 1.26	59.15 ± 1.17
Logistic	70.19 ± 1.54	76.50 ± 6.71	73.54 ± 8.86	68.80 ± 10.95	40.00 ± 11.45	59.19 ± 1.93	58.31 ± 2.39
1-NN CV	61.29 ± 1.91	81.34 ± 1.86	71.33 ± 5.22	72.89 ± 6.77	58.13 ± 1.42	63.12 ± 1.17	60.68 ± 1.84
5-NN CV	70.25 ± 1.35	79.47 ± 1.89	71.69 ± 7.92	71.49 ± 4.99	50.39 ± 3.65	62.26 ± 2.78	59.72 ± 1.75
KITE	73.86 ± 1.72	72.21 ± 3.33	80.29 ± 0.87	86.48 ± 2.23	39.52 ± 2.15	64.84 ± 1.88	67.90 ± 0.70

Table 6: The results of all the methods for each target dataset.

A.4 Discussion on Limitations and Extensions

This paper addresses an important problem in machine learning, and we believe the proposed method should not raise any ethical considerations. We discuss limitations and possible extensions below,

Computational Resources. In general, transferability estimation methods prefer to select deeper model. This is a desirable property if the computational resource is not a bottleneck. In practice, the users may not have enough resources. Thus, it would be interesting to take the actual resources the users have into account for transferability estimation.

Downstream Tasks. Currently, we only consider classification as the downstream task. It would be interesting to extend KITE for downstream tasks such as instance segmentation, semantic segmentation and object detection. To accurately estimate the fine-tuning performance of pre-trained models would greatly improve the performance of these tasks.