Harmonizing class uniformity and separability for transferability estimation

Yuhe Ding, Bo Jiang, Lijun Sheng, Aihua Zheng, Jian Liang

PII: S0031-3203(26)00862-9
DOI: https://doi.org/10.1016/j.patcog.2026.113897
Reference: PR 113897

To appear in: *Pattern Recognition*

Received date : 8 April 2025
Revised date : 22 January 2026
Accepted date : 28 April 2026

Please cite this article as: Y. Ding, B. Jiang, L. Sheng et al., Harmonizing class uniformity and separability for transferability estimation, *Pattern Recognition* (2026), doi: https://doi.org/10.1016/j.patcog.2026.113897.

Harmonizing Class Uniformity and Separability for Transferability Estimation

Yuhe Ding[a], Bo Jiang[a], Lijun Sheng[d,b], Aihua Zheng[c], Jian Liang[d,e,*]

[a]School of Computer Science and Technology Anhui University
[b]University of Science and Technology of China
[c]School of Artificial Intelligence Anhui University
[d]NLPR \& MAIS Institute of Automation Chinese Academy of Sciences
[e]School of Artificial Intelligence University of Chinese Academy of Sciences

Abstract

Transferability estimation aims to provide heuristics for quantifying how suitable a pre-trained model is for a specific downstream task, without fine-tuning them all. Existing methods embed target datasets into the feature space defined by the pre-trained model, guided by the common intuition that an ideal initial feature space has a clear decision boundary. However, these methods largely focus on separability and insufficiently account for class distribution bias inherited from the source domain and imprinted on the feature manifold, which can critically affect downstream fine-tuning performance. This paper tackles this problem and proposes a simple yet effective transferability estimation method, termed **H**arm**O**nizing **C**lass **U**niformity and **S**eparability (HOCUS). HOCUS includes a baseline class separability score and a novel class uniformity score, formulated as the entropy of the class distribution overlap matrix. The proposed uniformity score is flexible and can be easily integrated as a plug-and-play module into existing methods. We investigate our method on a variety of pre-trained classification models across different network architectures, source datasets, and training loss functions. Experimental results demonstrate that HOCUS achieves state-of-the-art performance in terms of Pearson and Kendall correlation across diverse model zoos, and that the class proposed uniformity score consistently enhances existing methods. Code will be released at https://github.com/YuheD/HOCUS.

*Corresponding author.

Keywords: Transferability Estimation; Transfer Learning; Model Evaluation

1. Introduction

Transfer learning has evolved into a mature field in recent years. The “pre-training then fine-tuning" has become a standard training paradigm for numerous tasks in the realm of deep learning and diverse repositories of pre-trained models, known as model zoos, are established[1]. These models are constructed through combinations of diverse network architectures, source datasets, and loss functions. This naturally raises a fundamental question: Given a specific downstream task, which pre-trained model should be selected as the most suitable starting point?

A naive strategy is to fine-tune all candidate models and select the one that achieves the best performance. However, for large-scale target datasets, this approach is prohibitively expensive in both time and computational resources. To address this challenge, transferability estimation has been proposed [1, 2, 3]. The goal is to design a metric that predicts how well a pre-trained model will perform on a target dataset without requiring exhaustive fine-tuning. An effective metric should be task-adaptive and exhibit strong correlation with the downstream fine-tuning performance. A widely adopted intuition is that the feature space induced by a well-trained model tends to form a clear decision boundary [4]. Building on this intuition, many existing transferability metrics [5, 6, 2] quantify class separability, often in combination with perspectives such as informativeness [2] or uncertainty [7]. The central idea is to measure how far different classes are separated in the feature space, typically through the sum or average of inter-class distances.

However, this intuition has a critical limitation that undermines the effectiveness of current approaches. Consider the illustrative example in Fig. 1: although the feature spaces in (a) and (b) exhibit similar levels of class separability, the model corresponding to (a) is biased toward the purple class. Solely relying on inter-class separability would incorrectly favor (a) as the best model. This example highlights a fundamental

[1] pytorch.org/hub; docs.openvino.ai/; tfhub.dev/

flaw: existing methods fail to account for class-specific bias. When a candidate model is biased toward head classes in the target dataset, it may be assigned a higher transferability score. In contrast, a model with stronger generalization ability should not only ensure clear separation between classes but also maintain fairness across all classes [4].

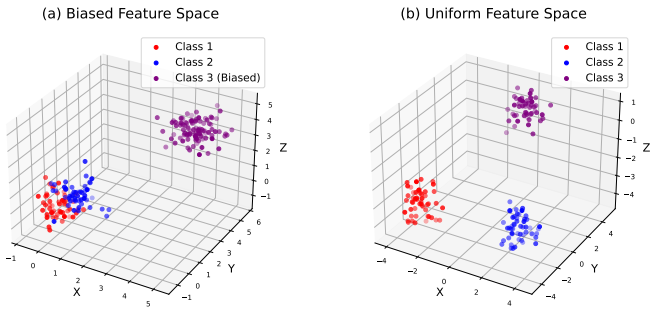

Figure 1: Two types of feature spaces with similar class separability scores. Each point represents a sample in the feature space, with different colors denoting distinct classes. (a) exhibits bias towards the purple class, while (b) maintains uniformity.

As a result, we propose a simple yet effective method termed **H**arm**O**nizing **C**lass **U**niformity and **S**eparability (HOCUS). HOCUS consists of a class separability score and a novel class uniformity score. Specifically, the class separability score is defined as the magnitude of between-class covariance compared to within-class covariance, serving as our baseline. To address the overlooked issue of class bias, we further design the class uniformity score (CU), formulated as the entropy of the class distribution overlapping matrix. A higher CU indicates that class distributions are more evenly spread in the feature space, reflecting greater fairness of the pre-trained model across all classes. HOCUS therefore provides a more comprehensive transferability metric by harmonizing separability with uniformity. Importantly, CU is a simple, flexible, and plug-and-play module: it can be seamlessly integrated into existing methods to enhance their performance, particularly for those that do not explicitly account for class uniformity.

Overall, our main contributions are summarized as follows:

- We introduce a simple yet effective transferability estimation framework termed **H**arm**O**nizing **C**lass **U**niformity and **S**eparability (HOCUS), including a basic class separability score and an additional class uniformity score.
- We propose a novel class uniformity score (CU), formulated as the entropy of the class distribution overlapping matrix. CU is flexible and can be easily integrated as a plug-and-play score into existing methods.
- To validate the effectiveness and generality of HOCUS and CU, we conduct experiments on both image classification and semantic segmentation tasks. We also consider various model zoos involving multiple model architectures, multiple loss functions, and multi-source datasets. Experimental results demonstrate that HOCUS yields state-of-the-art results for transferability estimation.

2. Related works

2.1. Transferability Estimation

As an important problem in transfer learning [8] and model evaluation [9, 10], transferability estimation facilitates model selection, allowing developers to choose models that are well-suited to specific target tasks without needing extensive domain-specific data. This is especially crucial in real-world scenarios where retraining models on vast, diverse datasets for every new application is often impractical or costly. In areas like healthcare, natural language processing, and autonomous systems, accurate transferability estimation ensures that models retain their accuracy and reliability when faced with new conditions, thereby enhancing their utility and impact. There has been an increasing amount of research in the field of transferability estimation [1, 11].

Existing methods could be roughly divided into two main types, *i.e.*, information theory-based methods and feature analysis-based methods. The information theory-based methods [12, 3] usually combine with the Bayesian theory, to measure the domain gap from a probabilistic perspective, or the informativeness of feature matrix. LEEP [12] is the classification performance on the Expected Empirical Predictor (EEP); Based on LEEP, Agostinelli *et al* [13] design four metrics, *i.e.*, MS-LEEP, E-LEEP,

IoU-LEEP, and SoftIoU-LEEP, for variant settings. NCE [3] considers the conditional entropy between the label assignments of the source and target tasks. LogME [14] is the maximum value of label evidence (marginalized likelihood) given extracted features.

Feature analysis-based methods [2, 5] analyze the structure in the features of pre-trained models on the target dataset. H-score [2] considers between-class variance and feature redundancy. It is an information-theoretic metric that estimates transferability by measuring how much useful information the source features retain for the target task, specifically balancing the discriminative power (via between-class covariance) against redundancy (via feature correlations). The score is computed as a ratio involving trace terms of covariance matrices. GBC [5] is the summation of the pairwise class separability using the Bhattacharyya coefficient. It assumes Gaussian class-conditional distributions in the feature space and quantifies the overlap between every pair of classes via the Bhattacharyya distance, with lower overlap (higher separability) indicating better transferability. NCTI [6] is inspired by the Neural Collapse [4], consisting of several components related to Neural Collapse. It quantifies how close the pre-trained model's feature representations on the target dataset are to the ideal Neural Collapse state. SFDA [15] leverages a self-challenging Fisher space, which captures the intrinsic characteristics of model representations. By evaluating the class separation in this space, the framework identifies how different models can transfer knowledge to new tasks. DISCO [16] analyzes the singular value decomposition of features, investigates different spectral components, and observes that they possess distinct transferability. SA [7] proposes an enhancement method, which enhances existing methods by introducing feature space perturbation to evaluate the robustness of feature for transferability estimation. PEFTDiff [17] introduces a diffusion-guided approach specifically tailored for selecting Parameter-Efficient Fine-Tuning (PEFT) techniques applied to a shared backbone. Unlike prior feature analysis methods that often rely on linear assumptions or simplified feature structures, PEFTDiff models the nonlinear geometric relationships in PEFT features using diffusion maps constructed from Gaussian RBF kernels and k-NN sparsification.

In summary, most feature analysis-based transferability estimation methods pri-

marily focus on quantifying class separability in the feature space, positing that greater inter-class separation indicates stronger transferability to the downstream task. While NCTI and our proposed HOCUS method are both inspired by the Neural Collapse (NC) phenomenon [4], NCTI adopts a relatively strict evaluation of how closely the pre-trained features on the target dataset approximate the full set of NC properties. In contrast, NC primarily emerges in well-trained models after sufficient optimization, whereas pre-trained models applied to new tasks often exhibit pronounced intra-class dispersion at initialization. Our method better suits the selection of source models by explicitly accounting for the dispersed within-class structure before fine-tuning.

Compared with information theory-based approaches, feature analysis-based methods are generally more convenient and intuitive. These methods operate directly on the representations of pre-trained models, thus avoiding additional probabilistic assumptions and complex Bayesian formulations. Moreover, feature-based metrics can capture fine-grained structural properties of features, such as class separability, redundancy, and robustness, which are closely aligned with the essence of transferability. Our method is a typical feature structure-based method, and compared to existing methods, we are the first to consider the fairness of pre-trained models towards target classes.

2.2. *Model Selection*

Based on the objective, there are many tasks to solve the model selection problem. Task transferability [18], also known as task similarity, aims to explore the relationship among visual tasks and offers a principled approach to identify redundancies across tasks. This topic is typically significant in multi-task learning and meta-learning problems. In these problems, highly similar tasks can often be jointly learned, leading to a decrease in the need for annotation and improved performance. Generalization Gap Prediction [19] methods predict the difference between the accuracy of the training data and unseen test data from the same distribution, *i.e.*, generalization gap. The generalization gap represents the model's ability to generalize from the training data to new, unseen data, constituting a vital aspect in the evaluation and enhancement of machine learning models. Out-of-distribution error prediction [20] involves assessing the model's performance using a test set that includes OOD data. It evaluates the general-

ization performance of a model within the same task across different data distributions, without involving transfer learning training for downstream tasks. Recently, vision-language model selection [21] leverage the class names only to select a generalization model for downstream tasks, it predict the zero-shot classification performance of the candidate vision language models on unknown datasets.

3. Method

3.1. Problem Formulation

We consider a K-way classification problem on a target dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where n denotes the total number of labeled samples and n_k represents the number of samples in the k-th class. In addition, we are given a pre-trained model zoo $\{\phi_m\}_{m=1}^{M}$ containing M candidate models.

The objective of transferability estimation is to assign each model ϕ_m a metric score T_m, such that the scores $\{T_m\}_{m=1}^{M}$ exhibit strong correlation with the ground-truth transfer performance, *i.e.*, the test accuracy of ϕ_m after fine-tuning on D.

3.2. Motivation

Most existing transferability estimation methods rely heavily on the intuition that a well-trained model induces a feature space with clear decision boundaries. Based on this assumption, they typically define a transferability score by measuring class separability, such as inter-class distances or between-class covariance [2, 5, 6]. While this perspective provides a useful baseline, it overlooks a critical aspect: the potential bias of pre-trained models towards specific classes. As illustrated in Fig. 1 (a), the purple class dominates the feature space, indicating a bias that could mislead separability-based metrics, whereas Fig. 1 (b) shows a more balanced distribution across classes. A model with stronger generalization ability should not only ensure clear separation between classes but also maintain fairness across all classes. This is not merely an intuition, but a key insight revealed by the Neural Collapse phenomenon [4]. We first introduce the phenomenon of neural collapse, and then present our method, HOCUS, which explicitly integrates both class separability and uniformity into a unified transferability estimation framework.

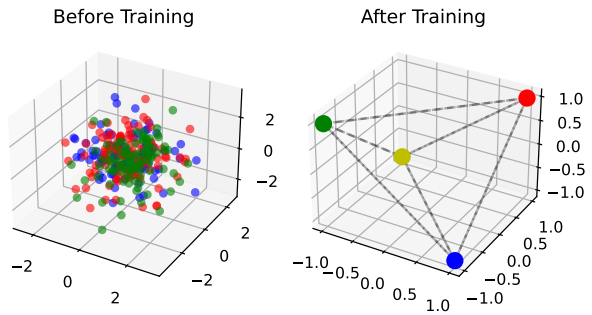

Figure 2: Illustration of Neural Collapse.

Neural Collapse (NC) is a pervasive inductive bias in the terminal phase of training (TPT) [4]. TPT begins at the epoch where the training error first vanishes, which is a sign of the completion of model training. Neural Collapse is characterized by four manifestations in the classifier and last-layer features: (NC1) the within-class variation collapses to zero; (NC2) the class means converge to Simplex Equiangular Tight Frame (ETF); (NC3) the class means and the weights of linear classifiers converge to each other; (NC4) the classifier converges to the nearest class-center classifier. These manifestations suggest models are learning maximally separable features between classes, which can be simplified as three properties: between-class separability, within-class compactness, and the equiangularity between each pair of classes.

The convergence of models towards NC usually results in the improvement of out-of-sample model performance and robustness to adversarial examples [4]. However, this commendable property generally occurs in the well-trained models, *i.e.*, fine-tuned models, rather than the pre-trained models. We further explore the relationship of NC between the pre-trained models and their corresponding fine-tuned models. To be specific, based on a rough metric of NC [22], we fine-tune several heterogeneous models pre-trained on ImageNet on two different target datasets, and track the changes in their NC scores. As shown in Fig. 3, we find that the NC score ranking in these pre-trained models remains mostly consistent during fine-tuning. Therefore, ranking the pre-trained models by their initial level of Neural Collapse might be an available solution.

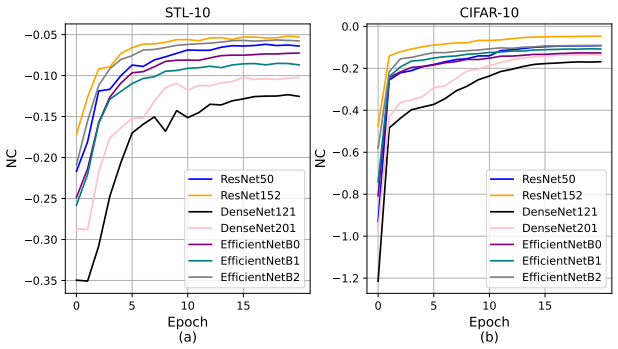

Figure 3: Observation of Neural Collapse during model fine-tuning on (a) STL-10 and (b) CIFAR-10, showing that initial NC rankings in pre-trained models persist, motivating our NC-inspired HOCUS metric.

As a result, we propose HarmOnizing Class Uniformity and Separability (HOCUS), a simple yet effective framework that integrates two complementary factors: 1) Class Separability, which captures the degree of inter-class discrimination in the feature space; and 2) Class Uniformity, a novel metric designed to quantify the fairness of pre-trained models by measuring how evenly class distributions are represented in the feature space. By jointly considering separability and uniformity, HOCUS provides a more comprehensive evaluation of the extent of Neural Collapse, while simultaneously serving as a reliable indicator of model transferability.

3.3. Harmonizing Class Uniformity and Separability (HOCUS)

We propose an overall method termed **H**arm**O**nizing **C**lass **U**niformity and **S**eparability (HOCUS), to measure both class uniformity and separability of the feature space defined by the pre-trained models. Specifically, the HOCUS score T_m for m-th candidate model, consists of two terms, class separability term S_m, and class uniformity term U_m. For m-th pre-trained model ϕ_m, the corresponding HOCUS score T_m is formulated as the sum of the normalized version of S_m and U_m,

$$T_m = \tilde{S}_m + \tilde{U}_m,\ \tilde{S}_m = \frac{S_m - S_{min}}{S_{max} - S_{min}}, \tilde{U}_m = \frac{U_m - U_{min}}{U_{max} - U_{min}}, \tag{1}$$

where S_m and U_m are the class separability and class uniformity score of the m-th pre-trained model ϕ_m, respectively. $\{S_m\}_{m=1}^{M}$ and $\{U_m\}_{m=1}^{M}$ are obtained from M pre-trained

models, U_{max} and S_{max} are the maximum scores, U_{min} and S_{min} are the minimum scores in $\{U_m\}_{m=1}^{M}$ and $\{S_m\}_{m=1}^{M}$. A higher HOCUS score T_m indicates that the model's feature space excels in both class separability and uniformity, thereby possessing greater transferability. Then we these two scores, *i.e.*, class separability and class uniformity, respectively. For clarity, we omit subscript m, since the procedure is identical for all models.

3.4. Class Separability Revisited

Given a feature matrix, classical Linear Discriminant Analysis (LDA) aims to find an optimal transformation such that the class structure of the original high-dimensional space is preserved in the low-dimensional space. In LDA, the quality of the feature structure is considered to be high if each class is tightly grouped and well-separated from the others. Maximizing the magnitude of between-class covariance compared to within-class covariance is employed as an optimization goal of LDA. Specifically, given the global mean $\boldsymbol{h}_G = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{h}_i$ and the class mean $\overline{\boldsymbol{h}}_k = \frac{1}{n_k}\sum_{i=1}^{n}\mathbb{1}(y_i = k)\boldsymbol{h}_i$, where $\mathbb{1}(\cdot)$ denotes the indicator function, the class separability score S can be formulated as,

$$
\begin{aligned}
S &= -\frac{1}{K}\operatorname{trace}\left(\boldsymbol{\Sigma}_W\boldsymbol{\Sigma}_B^{\dagger}\right),\\
\boldsymbol{\Sigma}_W &= \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{n}\frac{1}{n_k}\mathbb{1}(y_i = k)\left(\boldsymbol{h}_i - \overline{\boldsymbol{h}}_k\right)\left(\boldsymbol{h}_i - \overline{\boldsymbol{h}}_k\right)^{\top},\\
\boldsymbol{\Sigma}_B &= \frac{1}{K}\sum_{k=1}^{K}\left(\overline{\boldsymbol{h}}_k - \boldsymbol{h}_G\right)\left(\overline{\boldsymbol{h}}_k - \boldsymbol{h}_G\right)^{\top},
\end{aligned}
\tag{2}
$$

where K is the number of classes, $\boldsymbol{\Sigma}_W$ is the within-class covariance and $\boldsymbol{\Sigma}_B^{\dagger}$ is the pseudo inverse of between-class covariance $\boldsymbol{\Sigma}_B$. If the between-class covariance is much larger than the within-class covariance, it suggests that the classes are well-separated in the feature space, making it easier to discriminate between them. Conversely, if the within-class covariance dominates, it indicates that the groups are more spread out and less distinguishable from each other. As a result, the relative magnitude S of between-class covariance and within-class covariance is usually seen as a straightforward principle for class separability.

3.5. Class Uniformity

Consider a common scenario where the classification difficulty varies significantly across different classes in the label space. Features of simple classes are typically well-clustered, whereas features of difficult classes tend to be easily confusing in the feature space. Such an initial model fails to learn discriminative features for the difficult classes. An ideal pre-trained model should not exhibit any bias towards specific classes. We propose a novel class uniformity score to evaluate whether different classes are distributed fairly within the feature space.

To better depict the relationships between different classes, we first model each target class as a Gaussian distribution $\mathcal{N}(\overline{\boldsymbol{h}}_k, \boldsymbol{\Sigma}_k)$. $\boldsymbol{\Sigma}_k$ is the within-class covariance, which is defined as,

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^{n} \mathbb{1}(y_i = k) \left(\boldsymbol{h}_i - \overline{\boldsymbol{h}}_k\right) \left(\boldsymbol{h}_i - \overline{\boldsymbol{h}}_k\right)^{\top}, \tag{3}$$

where $\overline{\boldsymbol{h}}_k$ is the k-th class mean defined in Eq. (2). This step is non-trivial since the feature space of pre-trained models is scattered before fine-tuning, making it difficult to represent an entire class with just one or a few prototype vectors. Utilizing Gaussian distributions to represent the features of each class is a straightforward choice, as it allows for modeling both the location and the spread of the feature distribution simultaneously. Next, we calculate the overlap matrix $B \in \mathbb{R}^{K \times K}$, where each element represents the degree of overlap between distributions of each pair of classes. Here we employ the Bhattacharyya coefficient as the overlapping metric, as it considers both the mean and covariance and has a closed-form solution when applied between Gaussian distributions. Specifically, Bhattacharyya distance D between class k_i and k_j is calculated as follows,

$$D\left(k_i, k_j\right) = \frac{1}{8} \left(\overline{\boldsymbol{h}}_{k_i} - \overline{\boldsymbol{h}}_{k_j}\right)^{\top} \boldsymbol{\Sigma}^{-1} \left(\overline{\boldsymbol{h}}_{k_i} - \overline{\boldsymbol{h}}_{k_j}\right) + \frac{1}{2} \ln \frac{|\boldsymbol{\Sigma}|}{\sqrt{\left|\boldsymbol{\Sigma}_{k_i}\right| \left|\boldsymbol{\Sigma}_{k_j}\right|}}, \tag{4}$$

where $\boldsymbol{\Sigma} = \frac{1}{2}\left(\boldsymbol{\Sigma}_{k_i} + \boldsymbol{\Sigma}_{k_j}\right)$, $|\cdot|$ denotes determinant. Then, the Bhattacharyya coefficient is defined as $B\left(k_i, k_j\right) = exp - D\left(k_i, k_j\right)$, which indicates the overlap between different class distributions. We can thus obtain the overlap matrix $B \in \mathbb{R}^{K \times K}$, where the value at the i-th row and j-th column is $B\left(k_i, k_j\right)$. To highlight the difference between nearby

classes and far-away classes, we first convert the overlaps between classes into a probabilistic distribution P by using temperature scaling and softmax function. Then, we calculate the entropy for each row of the overlap matrix and define the class uniformity score U as,

$$U = -\frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{K} P_{ij}\log P_{ij}, \quad \text{where } P_{ij} = \frac{\exp\left(B\left(k_i,k_j\right)/t\right)}{\sum_{j'}\exp\left(B\left(k_i,k_{j'}\right)/t\right)}, \tag{5}$$

where t is the temperature in softmax function. Note that, when each row of P approaches a uniform distribution, the class uniformity score U reaches its maximum value, which indicates that any class distribution has a similar overlap with the distributions of other classes. It suggests that the pre-trained model is fair and exhibits no bias towards specific classes. HOCUS does not require any training or optimization, and its computational complexity is $O(n)$. We provide the detailed computation process in Algorithm 1.

Algorithm 1 Algorithm of the proposed HOCUS.

Input: A Model Zoo $\{\phi_m\}_{m=1}^{M}$ with M pre-trained models; target dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, with a total of n labeled samples, and there are n_k samples in the k-th class;

1: **repeat**
2: Given a pre-trained model ϕ_m, obtain the last-layer features $\{\boldsymbol{h}_i\}_{i=1}^{n}$ on $D = \{(x_i, y_i)\}_{i=1}^{n}$;
3: Calculate the class separability score S_m for ϕ_m by Eq. (2);
4: Calculate the class uniformity score U_m for ϕ_m by Eq. (5);
5: **until** Obtain $\{S_m\}_{m=1}^{M}$ and $\{U_m\}_{m=1}^{M}$ for $\{\phi_m\}_{m=1}^{M}$;
6: Rescale $\{S_m\}_{m=1}^{M}$ and $\{U_m\}_{m=1}^{M}$, and obtain the HOCUS score $\{T_m\}_{m=1}^{M}$ by Eq. (1).

Output: Transferability ranking of pre-trained models.

4. Experiments

In this section, we conduct a series of evaluations to assess the effectiveness of our proposed method, HOCUS, on various tasks. We begin with image classification, us-

ing heterogeneous model zoos with both single-source and multiple-source pre-trained models across several target datasets. We further evaluate HOCUS within a homogeneous model zoo, where models are trained with multiple sources and various loss functions. Additionally, we extended our experiments to model selection, semantic segmentation and text classification tasks, Vision transformer model zoo, class-imbalanced downstream datasets, and self-supervised algorithms, to explore HOCUS's generalizability across diverse settings.

Baseline Methods. In all the experiments, we compare our method with several state-of-the-art methods of various types[2]: LEEP (ICML, 2020) [12] and NCE (ICCV, 2019) [3], which are based on the joint distribution of source and target; H-score (ICIP, 2019) [2], GBC (CVPR, 2022) [5] and NCTI (ICCV, 2023) [6], which are based on the feature structure; LogME (ICML, 2021) [14], which is based on the maximum value of label evidence; SFDA (ECCV, 2022) [15], which is based on Fisher discriminant analysis; PEFTDiff (ICCV2025) [17], which is diffusion-guided.

Metrics. The coefficient between our metric and the fine-tuned accuracy is measured by Kendall rank correlation τ, which is usually used to measure non-linear, hierarchical, or sequential relationships, and Pearson correlation ρ, which is used for measuring linear relationships. These two are the basic metrics to measure the overall effectiveness of the transferability estimation methods, Besides, we additionally give the result of Top-5 Recall R_5, which measures the overlap between the five highest predicted models and the five actual optimal models,

$$R_5 = \frac{|F_5|}{5}, \quad F_5 = I(\mathcal{A}^5) \cap I(\mathcal{T}^5), \tag{6}$$

where $I(\cdot)$ is the index set, $\mathcal{A}^5$ and $\mathcal{S}^5$ are top-5 values in the ground truth $\mathcal{A} = \{a_m\}_{m=1}^M$ and the predicted scores $\mathcal{T} = \{T_m\}_{m=1}^M$, $\|F_5\|$ is the length of F_5.

Training Details. We provide the training details of image classification tasks below. Our settings follow existing works [12, 5]. For the training of pre-trained models on

[2]LEEP, NCE, LogME: github.com/thuml/LogME; H-score: git.io/J1WOr; NCTI: github.com/BUserName/NCTI; SFDA: github.com/TencentARC/SFDA; PEFTDiff: github.com/prafful-kumar/PEFT_SELECTION; GBC is implemented by us.

different source datasets except for ImageNet, we train the model pre-trained on ImgaeNet for 100 epochs, using an SGD optimizer with a learning rate of 0.01, and a batch size of 64. The pre-trained models on ImageNet are directly provided by Pytorch Model Hub[3]. For the model fine-tuning, we use the best parameters, the temperature t in Eq. (5) is empirically set to 0.05. Our experiments are conducted using the PyTorch framework on a 24G NVIDIA Geforce RTX 3090 GPU. We fine-tune the pre-trained models three times with seeds 0, 1, and 2, obtaining three results for each model. The maximum standard deviation was 0.86, demonstrating the stability of the training. We take the average results from the three seeds as the ground truth.

4.1. Image Classification: Heterogeneous Model Zoo with a Single Source

Experiment Setup. We construct a model zoo with 15 models pre-trained on ImageNet [23] across 5 architecture families: ResNet50, ResNet101, ResNet152 [24], DenseNet121, DenseNet169, DenseNet201 [25], MobileNetV1, MobileNetV2, MobileNetV3 [26], EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3 [27], VGG16, and VGG19 [28]. We use 7 standard image classification datasets as the target datasets: basic image recognition datasets CIFAR-10 [29], CIFAR-100 [29], and STL-10 [30]; animal dataset Oxford Pets [31] and CUB [32]; traffic sign dataset GTSRB [33]; describable textures dataset DTD [34]; large-scale and real world datasets SUN397 [35] and Food101 [36].

Quantitative Results. Table 1 summarizes the evaluation of different model scoring methods on heterogeneous model zoos using a single source. We report three metrics: Kendall's τ, Pearson's ρ, and Top-5 recall (R_5), to assess the correlation between predicted and actual model performances. Across datasets, HOCUS consistently achieves competitive performance. It attains the highest Kendall's τ on CUB, DTD, and STL-10, and ranks among the top two on most other datasets, demonstrating stable ranking ability. In terms of Pearson's ρ, HOCUS performs particularly well on CIFAR-10, indicating strong linear correlation with the actual accuracies. For Top-5 recall, HOCUS achieves a high average recall, reflecting its ability to reliably identify the suitable

[3] pytorch.org/hub

Table 1: Heterogeneous model zoo with a single source. **Bold** is the best result, underline is the second-best. Avg. is the average results on all datasets, and Avg.* is the average results except two large-scale real-world datasets SUN397 and Food101.

Target	Method								
	LEEP	NCE	LogME	H-score	GBC	NCTI	SFDA	PEFTDiff	HOCUS
Kendall (τ)									
CIFAR-10	0.62	0.81	0.75	0.71	0.79	0.77	**0.84**	0.68	0.81
CIFAR-100	0.70	0.85	0.52	0.60	**0.89**	0.64	0.63	0.81	0.83
Oxford Pets	-0.13	**0.82**	0.57	0.32	0.34	0.53	0.33	0.41	0.39
CUB	-0.34	-0.19	0.06	0.23	0.23	-0.31	-0.11	0.23	**0.33**
GTSRB	0.20	0.07	-0.37	0.10	0.10	-0.05	**0.30**	0.01	0.10
DTD	-0.02	0.54	0.29	0.46	0.52	**0.56**	0.54	**0.56**	**0.56**
STL-10	-0.24	0.83	0.87	0.54	0.83	0.79	0.69	0.71	**0.90**
SUN397	-0.26	**0.30**	0.20	-0.12	0.22	0.18	-0.39	0.20	0.24
Food101	0.22	0.14	0.26	0.24	**0.60**	0.62	0.15	0.37	0.58
Avg.*	0.11	0.53	0.38	0.42	0.51	0.51	0.46	0.48	**0.56**
Avg.	0.08	0.46	0.35	0.34	0.49	0.48	0.33	0.44	**0.53**
Pearson (ρ)									
CIFAR-10	0.57	0.87	0.76	0.82	0.87	0.83	0.73	0.84	**0.89**
CIFAR-100	0.69	**0.93**	0.67	0.56	**0.93**	0.70	0.63	0.84	0.85
Oxford Pets	-0.34	**0.93**	0.45	0.32	0.59	0.70	0.41	0.77	0.64
CUB	-0.38	-0.03	0.12	**0.57**	**0.57**	0.02	0.21	0.39	0.39
GTSRB	-0.28	0.15	-0.37	0.18	0.10	-0.05	**0.30**	0.04	-0.05
DTD	-0.15	0.58	0.29	0.48	**0.85**	0.71	0.65	0.62	0.71
STL-10	-0.30	**0.92**	0.61	0.65	0.83	0.79	0.50	0.86	0.90
SUN397	-0.32	**0.38**	0.23	-0.33	0.37	0.35	-0.74	0.28	0.32
Food101	0.30	0.22	0.31	-0.01	0.64	**0.75**	0.19	0.44	0.73
Avg.*	0.06	**0.62**	0.30	0.38	0.61	0.61	0.48	**0.62**	**0.62**
Avg.	0.05	0.55	0.29	0.26	0.59	**0.60**	0.31	0.56	**0.60**
Top-5 Recall (R_5)									
CIFAR-10	0.60	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
CIFAR-100	0.80	0.80	0.80	0.80	0.80	0.80	1.00	0.80	0.80
Oxford Pets	0.20	0.80	0.80	0.60	0.60	0.80	0.60	0.60	0.60
CUB	0.20	0.00	0.00	0.60	0.20	0.60	0.60	0.20	0.40
GTSRB	0.60	0.60	0.40	0.60	0.20	0.40	0.80	0.60	0.40
DTD	0.40	0.80	0.60	0.60	0.80	0.60	0.60	0.60	0.60
STL-10	0.20	0.80	1.00	0.60	1.00	0.80	0.60	0.80	1.00
SUN397	0.40	0.40	0.20	0.00	0.40	0.40	0.00	0.40	0.40
Food101	0.60	0.4	0.6	0.40	0.80	0.80	0.40	0.80	0.60
Avg.*	0.43	0.66	0.63	0.66	0.63	0.69	**0.71**	0.63	0.66
Avg.	0.44	0.60	0.58	0.56	0.62	**0.67**	0.60	0.62	0.62

models. Notably, HOCUS maintains stable performance even on large-scale real-world datasets such as SUN397 and Food101, further validating its robustness across diverse downstream scenarios. Note that GBC also yields competitive results in all three metrics. GBC also delivers competitive results across all three metrics. As a measure based on between-class overlap, GBC shares conceptual similarities with HOCUS. However, by additionally accounting for class uniformity, HOCUS consistently outperforms GBC, demonstrating the benefit of jointly modeling inter-class separation and intra-class balance.

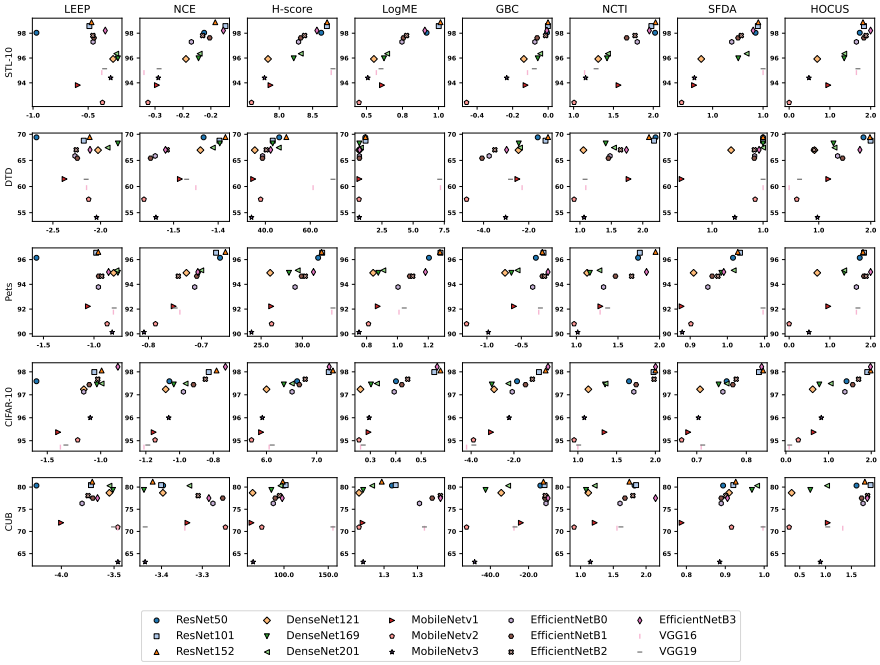

Figure 4: Qualitative results on the heterogeneous model zoo with a single source. For five various datasets, we show the visualized correlation between the accuracy of the fine-tuned model (Y-axis) and the transferability scores (X-axis) of LEEP, NCE, H-score, LogME, GBC, NCTI, SFDA and HOCUS.

Qualitative Results. We show the qualitative results in Fig. 4, *i.e.*, a correlation scatter figure between the fine-tuned accuracy and the transferability scores of the comparison methods, where the Y-axis is the fine-tuned accuracy, and the X-axis is the transferability score. Pre-trained models with higher fine-tuned accuracy should have higher

transferability scores. Therefore, methods where the scatter plot shows an increasing trend are considered superior. We do not achieve the best results in individual experiments, but we still exhibit an obvious increasing trend.

4.2. Image Classification: Heterogeneous Model Zoo with Multiple Sources

Experiment Setup. We construct a more complex model zoo in this experiment. Specifically, there are a total of 30 heterogeneous pre-trained models from 3 similar magnitude architectures (ResNet50, DenseNet121, and EfficientNetB2) pre-trained on 10 source datasets (CIFAR-10 [29], CIFAR-100 [29], CUB [32], Oxford Flowers [37], Stanford Cars [38], Country211 [39], SVHN [40], FGVC Aircraft [41]). These datasets encompass a wide range of image types, including animals, plants, digits, food, street, transportation, etc. We conduct the experiments on seven benchmark target datasets the same as Sec. 4.1.

Results. The experimental results, shown in Table 2, reveal that the proposed HOCUS method consistently achieves top performance across multiple target domains, as measured by the average τ and ρ. HOCUS demonstrates its effectiveness in complex, multi-source model zoo scenarios, outperforming other methods in both single- and multi-source environments. While NCE is effective in scenarios involving a single source, it performs less robustly when dealing with multiple sources. This may be attributed to the limitations of NCE in capturing the intricate relationships between multiple sources and target domains. In contrast, methods like GBC and NCTI, which leverage class separability and neural collapse principles, show competitive results, especially in the CUB and Oxford Pets datasets. This indicates that these methods are well-suited for certain types of data but may not fully utilize the high-dimensional feature representations that are advantageous in a multi-source context. PEFTDiff achieves competitive performance with an average Kendall's τ and Pearson's ρ of 0.40, matching or approaching several established baselines such as SFDA and GBC on certain datasets. Nevertheless, HOCUS consistently outperforms PEFTDiff and all other compared methods in terms of average correlation. The results underscore the potential of HOCUS for the model zoo with diverse sources, setting a new benchmark in both τ and ρ. This performance highlights the importance of class uniformity in navigating

the complexities of multi-source domain adaptation, making HOCUS a promising approach for future research and applications in similar settings.

Table 2: Heterogeneous model zoo with multiple sources.

Target	Method								
	LEEP	NCE	LogME	H-score	GBC	NCTI	SFDA	PEFTDiff	HOCUS
Kendall (τ)									
CIFAR-10	0.06	0.41	0.32	0.28	**0.45**	0.31	0.27	**0.45**	0.40
CIFAR-100	-0.10	0.20	0.04	-0.02	0.18	-0.01	-0.09	**0.25**	0.17
Oxford Pets	0.40	0.48	0.62	0.62	0.59	0.59	0.58	0.54	**0.63**
CUB	0.43	0.45	**0.66**	0.37	0.62	0.63	0.36	0.46	0.56
GTSRB	0.15	**0.16**	-0.16	-0.10	-0.05	-0.03	0.04	0.07	-0.05
DTD	0.09	0.44	0.41	0.21	0.52	0.52	0.18	0.59	**0.66**
STL-10	0.36	0.41	0.54	0.39	0.52	0.39	0.36	0.45	**0.57**
Avg.	0.20	0.36	0.35	0.25	0.40	0.34	0.24	0.40	**0.42**
Pearson (ρ)									
CIFAR-10	0.23	0.32	0.27	0.33	0.61	0.50	0.36	0.49	**0.64**
CIFAR-100	-0.01	0.05	-0.03	-0.03	**0.14**	0.04	-0.03	0.12	0.09
Oxford Pets	0.44	0.49	0.57	0.63	0.82	**0.85**	0.72	0.53	0.61
CUB	0.37	0.44	0.31	0.41	**0.82**	0.77	0.52	0.46	0.77
GTSRB	**0.36**	**0.36**	0.29	0.23	0.00	0.23	0.25	0.09	0.06
DTD	0.34	0.63	0.16	-0.07	0.78	0.73	0.25	0.50	**0.90**
STL-10	0.35	0.52	0.67	0.62	0.71	0.69	0.70	0.59	**0.84**
Avg.	0.30	0.40	0.32	0.30	0.55	0.54	0.40	0.40	**0.56**

4.3. Image Classification: Homogeneous Model Zoo with Multiple Sources and Loss Functions

Experiment Setup. We also construct a homogeneous model zoo, to comprehensively assess the capability of our method. There are a total of 21 ResNet50 models pretrained on 3 source datasets (CIFAR-10 [29], Oxford Pets [31] and CUB [32]) with 7 widely-employed loss functions [4] (cross entropy, label smoothing, MixUp [42], CutMix [43], Cutout [44], large margin softmax cross entropy [45], and Taylor softmax cross entropy [46]). We conduct the experiments on target datasets the same as Sec. 4.1.

Results. Table 3 presents the results for various methods within a homogeneous model

[4] github.com/fastai/fastai

Table 3: Homogeneous model zoo with multiple sources and loss functions.

Target	Method								
	LEEP	NCE	LogME	H-score	GBC	NCTI	SFDA	PEFTDiff	HOCUS
Kendall (τ)									
CIFAR-10	-0.07	-0.12	-0.05	-0.03	-0.10	-0.04	0.01	**0.10**	-0.06
CIFAR-100	-0.20	0.05	**0.29**	0.27	0.19	0.23	0.01	0.27	0.02
Oxford Pets	**0.50**	0.47	0.38	0.41	0.37	0.27	0.48	0.23	0.35
CUB	0.54	0.70	0.72	0.70	**0.83**	0.26	0.61	0.70	0.69
GTSRB	-0.08	-0.23	-0.16	-0.17	**0.24**	0.13	-0.16	0.05	0.19
DTD	-0.13	0.37	**0.65**	0.02	0.33	0.56	0.31	0.63	0.53
STL-10	-0.40	-0.25	0.42	0.63	0.04	0.21	**0.76**	0.00	0.58
Avg.	0.02	0.14	0.32	0.26	0.27	0.23	0.29	0.28	**0.33**
Pearson (ρ)									
CIFAR-10	-0.03	0.00	-0.11	-0.01	0.08	-0.10	-0.03	**0.20**	0.04
CIFAR-100	-0.74	-0.11	-0.05	-0.02	-0.01	-0.10	-0.17	**0.21**	-0.30
Oxford Pets	0.52	0.65	0.52	0.76	0.74	0.69	**0.80**	0.75	0.67
CUB	0.62	0.61	0.45	0.62	**0.83**	0.70	0.67	0.66	0.72
GTSRB	0.01	-0.18	-0.11	-0.06	0.37	0.32	-0.04	0.03	**0.38**
DTD	0.14	0.74	**0.93**	0.17	0.35	0.82	0.49	0.79	0.70
STL-10	-0.78	-0.54	0.44	0.71	-0.10	-0.01	**0.86**	-0.47	0.68
Avg.	-0.04	0.17	0.30	0.31	0.32	0.33	0.37	0.31	**0.41**

zoo, where multiple sources and loss functions are utilized. HOCUS emerges as the top performer in terms of average τ (0.33) and ρ (0.41), despite not having the highest score for every individual target dataset. This suggests that HOCUS maintains a strong overall performance and demonstrates robust generalization capabilities across different datasets. A closer look at the results shows that while methods like SFDA, LogME, PEFTDiff, and NCTI achieve competitive performance, particularly on specific datasets such as STL-10 and DTD, they do not consistently perform as well on other datasets. For example, SFDA shows a strong Pearson correlation on STL-10 (ρ = 0.86), highlighting its ability to capture certain domain characteristics effectively. However, these methods tend to perform less consistently when averaged across all datasets. In contrast, HOCUS provides a more balanced performance across both metrics, indicating its adaptability to a variety of datasets in the homogeneous model zoo setup. This balance suggests that HOCUS effectively leverages information from multiple sources and loss functions, enabling it to handle diverse data distributions with

relative ease. These findings reinforce HOCUS's versatility and adaptability in scenarios with uniform model structures but varied source data and loss functions. While other methods may excel on specific datasets, HOCUS stands out for its consistent and high average performance, making it a promising approach for applications requiring robust cross-domain generalization in a homogeneous model zoo with multiple sources and loss functions.

4.4. Model Selection

Table 4: Average accuracy (%) of the selected best model on (a) heterogeneous model zoo with a single source; (b) heterogeneous model zoo with multiple sources; (c) homogeneous model zoo with multiple sources and loss functions. Oracle refers to the accuracy of the actual best model in the model zoo.

Model Zoo	Target	Method								
		LEEP	NCE	LogME	H-score	GBC	NCTI	SFDA	HOCUS	Oracle
(a)	CIFAR-10	98.22	98.22	98.06	98.06	98.22	98.22	98.06	98.06	98.22
	CIFAR-100	88.37	88.37	88.66	88.66	88.66	88.66	88.66	88.66	88.66
	Oxford Pets	94.93	96.62	96.55	92.10	95.00	96.62	91.75	95.00	96.62
	CUB	63.12	70.95	77.51	70.73	77.51	77.51	71.02	80.45	81.22
	GTSRB	97.21	96.18	96.21	97.02	96.13	96.13	97.02	97.02	97.64
	DTD	68.25	69.47	59.79	61.38	69.47	69.42	69.42	68.83	69.47
	STL-10	95.98	98.58	98.89	95.14	98.89	98.89	95.14	98.21	98.89
	Avg.	86.58	88.34	87.95	86.16	89.13	89.35	87.30	**89.46**	90.10
(b)	CIFAR-10	96.83	96.83	96.83	96.83	96.83	96.83	96.83	96.83	97.51
	CIFAR-100	83.28	83.28	83.28	83.28	83.28	83.28	83.28	83.28	95.87
	Oxford Pets	93.98	93.98	93.98	94.66	93.58	93.58	94.66	93.58	94.66
	CUB	76.75	77.67	77.67	77.67	77.67	77.67	77.67	77.67	77.82
	GTSRB	97.47	97.05	97.05	97.05	97.05	97.05	97.05	97.05	97.50
	DTD	62.07	64.42	65.69	64.15	62.34	65.69	47.07	64.42	65.69
	STL-10	96.48	95.33	95.33	95.33	96.48	92.16	95.33	96.48	97.58
	Avg.	86.69	86.94	**87.12**	87.00	86.75	86.61	84.55	87.04	89.52
(c)	CIFAR-10	96.56	96.56	96.56	96.56	96.56	96.56	96.56	97.26	97.26
	CIFAR-100	80.38	83.86	84.46	84.46	84.46	84.34	84.46	84.46	84.46
	Oxford Pets	93.24	94.05	93.24	93.24	93.04	93.24	94.05	93.24	94.05
	CUB	77.86	77.86	76.96	76.96	78.43	76.96	77.86	80.31	80.31
	GTSRB	97.41	96.68	96.61	96.61	97.41	93.53	96.61	96.96	97.53
	DTD	62.88	63.62	64.42	63.62	63.40	63.62	51.22	64.42	64.47
	STL-10	92.34	95.23	95.15	95.15	94.61	95.23	95.15	95.15	95.80
	Avg.	85.81	86.84	86.77	86.66	86.84	86.21	85.13	**87.40**	87.70

Transferability estimation can be applied to various scenarios, such as in ensemble learning, multi-task learning, and multi-source domain adaptation, etc. Selecting the

best pre-trained model for downstream fine-tuning is a vital practice of transferability estimation. We present the average accuracy of the best models selected by HOCUS and existing methods, an indicator commonly used in the field of model validation in Table 4. Rows (a), (b), and (c) correspond to the average results in Table 1, 2, and 3, respectively. It reflects the average expected performance of these methods when applied to the model selection task. In (a) single-source heterogeneous model zoo, HOCUS achieves strong performance with an average accuracy of 89.46%, close to the Oracle benchmark of 90.10%. This result indicates that HOCUS is capable of effectively selecting models with high accuracy in single-source settings. Additionally, GBC and NCTI perform similarly, with average accuracies of 89.13% and 89.35%, respectively, highlighting their robustness and adaptability across a range of datasets. In (b) multi-source heterogeneous model zoo, LogME leads with an average accuracy of 87.12%, slightly outperforming both HOCUS (87.04%) and H-score (87.00%). This setup demonstrates that LogME excels in multi-source scenarios, suggesting it may better handle the complexities introduced by multiple-source distributions. However, HOCUS and H-score remain competitive, indicating their ability to generalize well even in more complex model zoo configurations. In (c) homogeneous model zoo with multiple sources and loss functions, HOCUS achieves an average accuracy of 87.40%, once again placing it near the top and only slightly below Oracle's 87.70%. Here, GBC and NCE also perform well, each with average accuracies of 86.84%.

4.5. Effectiveness Analysis

In this section, we explore the effectiveness of the component of HOCUS. Specifically, we conduct plug-and-play evaluation of class uniformity score, the ablation study of class separability score and class uniformity score, and the comparison with the Naive Equiangularity Metric. These experiments are conducted on three model zoo: (a) heterogeneous model zoo with a single source (Sec. 4.1); (b) heterogeneous model zoo with multiple sources (Sec. 4.2); (c) homogeneous model zoo with multiple sources and loss functions (Sec. 4.3), where the target dataset adopts the common parts of the three.

Plug-and-play Evaluation of Class Uniformity Score. Our class uniformity score

(CU) is a plug-and-play module, that can be integrated directly into existing methods. We construct diverse variants of HOCUS to validate the effectiveness of HOCUS and CU. Specifically, we replace $\tilde{S}$ in Eq. (1) with 5 representative methods that do not consider the model's bias toward target classes, to construct 5 integrated methods respectively. Besides, we also provide results using the class uniformity score (CU) and class separability score (CS), individually. All these experiments are conducted on the three model zoos described in Sec. 4.1, 4.2, and 4.3. The results are shown in Table 5. For most methods, the integration of CU leads to improvements. The improvement for GBC is not very significant, which could be because GBC implicitly considers class uniformity when modeling class distributions by taking into account the within-class variance. The class uniformity and CU score complement each other, usually resulting in a better performance compared to using each of them individually.

Table 5: Plug-and-play results on (a) heterogeneous model zoo with a single source (Sec. 4.1); (b) heterogeneous model zoo with multiple sources (Sec. 4.2); (c) homogeneous model zoo with multiple sources and loss functions (Sec. 4.3).

Method	(a)		(b)		(c)	
	τ (Avg.)	ρ (Avg.)	τ (Avg.)	ρ (Avg.)	τ (Avg.)	ρ (Avg.)
LEEP	0.11	0.06	0.20	0.30	0.02	-0.04
LEEP+CU	0.37 (+0.26)	0.48 (+0.42)	0.33 (+0.13)	0.40 (+0.10)	0.15 (+0.13)	0.16 (+0.20)
NCE	0.53	0.62	0.36	0.40	0.14	0.17
NCE+CU	0.59 (+0.06)	0.69 (+0.07)	0.41 (+0.05)	0.44 (+0.04)	0.28 (+0.14)	0.29 (+0.12)
LogME	0.38	0.30	0.35	0.32	0.32	0.30
LogME+CU	0.45 (+0.07)	0.47 (+0.17)	0.37 (+0.02)	0.40 (+0.08)	0.35 (+0.03)	0.42 (+0.12)
H-score	0.42	0.38	0.25	0.30	0.26	0.31
H-score+CU	0.46 (+0.04)	0.57 (+0.19)	0.36 (+0.11)	0.41 (+0.11)	0.31 (+0.05)	0.39 (+0.08)
GBC	0.51	0.61	0.40	0.55	0.27	0.32
GBC+CU	0.55 (+0.04)	0.64 (+0.03)	0.41 (+0.01)	0.50 (-0.05)	0.32 (+0.05)	0.35 (+0.03)
SFDA	0.46	0.48	0.27	0.48	0.29	0.37
SFDA+CU	0.58 (+0.12)	0.60 (+0.12)	0.46 (+0.19)	0.67 (+0.19)	0.35 (+0.06)	0.44 (+0.07)

Ablation Study. To validate the contribution of each component in our proposed metric, we conduct a comprehensive ablation study on the three benchmark settings: (a) heterogeneous model zoo with a single source, (b) heterogeneous model zoo with multiple sources, and (c) homogeneous model zoo with multiple sources and loss functions. The results are summarized in Table 6. Both CS and CU are effective individually. Using CU alone consistently yields strong performance. This suggests that the property of class uniformity is a highly reliable signal for predicting model transferability. The combination of CS and CU achieves the best or competitive performance across all benchmarks, demonstrating that the two components capture complementary aspects of transferability.

Table 6: Ablation study of class separability score (CS) and class uniformity score (CU) on (a) heterogeneous model zoo with a single source (Sec. 4.1); (b) heterogeneous model zoo with multiple sources (Sec. 4.2); (c) homogeneous model zoo with multiple sources and loss functions (Sec. 4.3).

CS	CU	(a)		(b)		(c)	
		τ (Avg.)	ρ (Avg.)	τ (Avg.)	ρ (Avg.)	τ (Avg.)	ρ (Avg.)
✓	×	0.49	0.52	0.30	0.38	0.33	0.28
×	✓	0.54	0.61	0.39	0.47	0.34	0.31
✓	✓	0.56	0.62	0.42	0.56	0.33	0.41

Comparison with the Naive Equiangularity Metric. HOCUS is inspired by Nueral Collapse, and the class uniformity score quantifies how evenly the class distributions are spread in the feature space, aligning with the equiangularity in NC2. This naturally raises an interesting question: Would directly using the naive NC2 metric [22, 4] , *i.e.*, the closeness between class means and a simplex ETF, perform better than our class uniformity score? Specifically, equiangularity can be estimated by,

$$\mathcal{NC}_2(\boldsymbol{W}) = \left\| \frac{\boldsymbol{W}\boldsymbol{W}^\top}{\|\boldsymbol{W}\boldsymbol{W}^\top\|_F} - \frac{1}{\sqrt{K-1}}\left(\boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\right)\right\|_F, \tag{7}$$

where $\boldsymbol{W} \in \mathbb{R}^{K\times d}$ is the weight of the classifier. This solution uses the weight $\boldsymbol{W}$ of the classifier as a representation of a class. In our task, this is actually an equiangularity metric for the unknown source dataset rather than the target dataset, since the model is pre-trained on the source dataset. Due to the self-duality between model weights

and class means [4], a naive solution is to replace $\boldsymbol{W}$ in the Eq. (7) with target class means matrix $\boldsymbol{A} = cat(\{\overline{\boldsymbol{h}}_k\}_{k=1}^{K}) \in \mathbb{R}^{K\times d}$, where $cat(\cdot)$ is the concatenation operation. As shown in Table 7, we conduct the comparison experiments of our class uniformity score and this naive solution on the heterogeneous model zoo with a single source (Sec. 4.1). CU is our class uniformity score, which is obviously superior to $\mathcal{NC}_2(\boldsymbol{A})$. The equiangularity in Neural Collapse essentially implies the maximum separability of class distributions in the feature spaces. When the within-class variance collapses to zero, each class mean can represent the corresponding entire class distribution. In the cases of a pre-trained model without fine-tuning, the within-class variance is large, hence the closeness between the class means and a simplex ETF cannot accurately measure the separability of class distributions.

Table 7: Comparison between our class uniformity score and the naive equiangularity metric on heterogeneous Model Zoo with a Single Source (Sec. 4.1).

	CU (Ours)	$\mathcal{NC}_2(\boldsymbol{A})$
τ (Avg.)	0.56	0.32
ρ (Avg.)	0.62	0.37

4.6. Sensitivity Analysis

We conduct sensitivity analysis on the temperature t in Eq. (5), and additionally validate the sensitivity on the class distribution distance metric and the score combination coefficient. The complete results are shown in Table 8.

Sensitivity of Temperature. The default value of temperature t is simply determined by calculating the average value of $B\left(k_i, k_j\right)$ in Eq. 5 during the experiments. We select an approximate number of the same magnitude. The role of t is to sharpen $B\left(k_i, k_j\right)$, ensuring reasonable results when calculating entropy. The default value t=0.05 remains consistent across all experiments, including various target datasets, different model zoos, and diverse tasks. The results are shown in rows named t in Table 8. The temperature parameter (t) shows minimal impact on the transferability estimation results across all three configurations. The τ and ρ values remain consistently stable across a range of t values, indicating that our method is robust to variations in temperature. This

Table 8: Sensitivity analysis of temperature, class distribution distance, and combination coefficient on (a) heterogeneous model zoo with a single source; (b) heterogeneous model zoo with multiple sources; (c) homogeneous model zoo with multiple sources and loss functions.

Variable		(a)		(b)		(c)	
		τ (Avg.)	ρ (Avg.)	τ (Avg.)	ρ (Avg.)	τ (Avg.)	ρ (Avg.)
t	0.010	0.56	0.59	0.40	0.49	0.36	0.43
	0.020	0.55	0.62	0.40	0.51	0.34	0.40
	0.050	0.56	0.62	0.42	0.56	0.33	0.41
	0.225	0.56	0.64	0.40	0.56	0.34	0.40
	0.500	0.58	0.65	0.40	0.57	0.35	0.41
Dist.	Bh	0.56	0.62	0.42	0.56	0.33	0.41
	Maha	0.51	0.58	0.37	0.45	0.33	0.45
λ	0.40	0.56	0.59	0.38	0.55	0.32	0.41
	0.70	0.55	0.61	0.39	0.55	0.34	0.40
	1.00	0.56	0.62	0.42	0.56	0.33	0.41
	1.30	0.55	0.64	0.38	0.53	0.34	0.39
	1.60	0.55	0.64	0.38	0.53	0.34	0.38

stability suggests that users can select a broad range of t values without significantly affecting performance, demonstrating that our method is not highly sensitive to this parameter.

Sensitivity of Class Distribution Distance Metric. Before fine-tuning, the feature space of models tends to be scattered, with significant variation in the radius (*i.e.*, covariance) of the Gaussian distributions of each class. Therefore, a coefficient that considers both the mean and covariance of each Gaussian distribution is crucial. Bhattacharyya coefficient is a simple and commonly used choice for this purpose. However, it is not the only choice, other metrics that take into account both the mean and covariance of the Gaussian distributions can be selected. For example, Mahalanobis distance, which is given by $D_{Maha}(k_i, k_j) = \sqrt{(\bar{h}_{k_i} - \bar{h}_{k_j})^T \Sigma^{-1} (\bar{h}_{k_i} - \bar{h}_{k_j})}$. We provide a comparison between the Mahalanobis distance version of HOCUS (Maha) and the Bhattacharyya distance version of HOCUS (Bh) in the rows named Dist. in Table 8. The method performs similarly when using either the Bhattacharyya or Mahalanobis distance metrics, as indicated by comparable τ and ρ values, particularly in configurations (b) and (c). While the Bhattacharyya distance slightly outperforms Mahalanobis

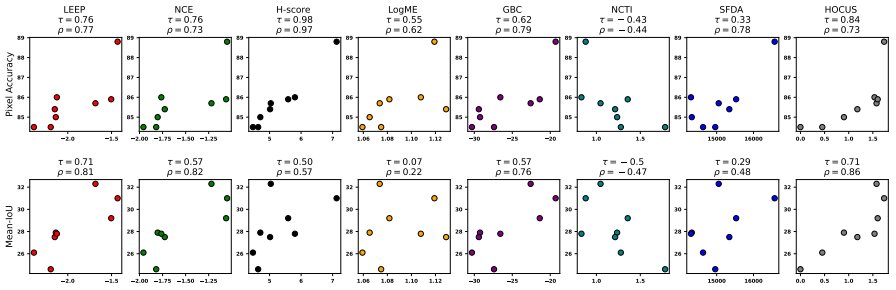

Figure 5: Quantitative and qualitative results on semantic segmentation model zoo. The X-axis represents the corresponding transferability score, and the Y-axis represents the fine-tuned pixel accuracy and mean IoU, respectively.

in most cases, the overall results are stable regardless of the choice of metric. This low sensitivity to the distance metric highlights the flexibility of our method, making it applicable across different class distribution metrics without compromising on effectiveness.

Sensitivity of Combination Coefficient. There are two components in HOCUS, which we combine with equal weights. In this section, we investigate the sensitivity of the HOCUS score to the combination weights of these two components. Specifically, we define the combination version of Eq. (1) as $\tilde{T} = \tilde{S} + \lambda\tilde{U}$. The combination coefficient λ is set to 1 by default, we take several different values around the default value, and the results are shown in rows named λ in Table 8. The combination coefficient (λ) has minimal impact on the performance, with τ and ρ values remaining stable over a broad range from 0.4 to 1.6. Moderate values around 1.0 provide optimal results, but even as λ varies, our method consistently achieves competitive results across all settings. This robustness to changes in λ further demonstrates the method's stability, allowing users to flexibly adjust the coefficient without a significant impact on transferability estimation outcomes.

4.7. Case Study: Semantic Segmentation

Experiment Setup. To validate the generalizability of our method, we also conduct experiments in the semantic segmentation scenario. Different from the standard back-

bone with classification head structure of classification tasks, the model architecture of segmentation tasks is diverse, so we evaluate the homogeneous models on segmentation tasks. We train 8 models on PSPNet [47] with ResNet50 backbone to construct our segmentation model zoo. These models are trained on 8 different source datasets: ADE20K [48], VOC [49], VOC Aug [49], SBU shadow [50], MSCOCO [51], LIP [52], kitti [53], and Camvid [54]. Note that We compare our method with the state-of-the-art methods on the standard segmentation target dataset CityScapes [55], following the open-source semantic segmentation benchmark [5].

Results. We present both quantitative and qualitative results in Fig. 5. In the scenario of pixel accuracy, most of these methods have a satisfactory result, while in the scenario of mean IoU, the performance of these methods has an obvious drop. Pixel accuracy is a metric on the pixel classification problem, while semantic segmentation is essentially a dense prediction problem. HOCUS obtains competitive results on pixel accuracy, and the best results on mean IoU. NCTI seems to have failed, possibly due to its strict design for each property of NC, while segmentation tasks have a larger output space and some differences from traditional classification tasks. LogME fails in mean IoU, while NCE and H-score also have a degree of decline. LEEP, GBC, and HOCUS yield similar results under both metrics, demonstrating the generalizability of these three methods in segmentation tasks. HOCUS is uniquely effective in accounting for class biases, which is crucial for semantic segmentation tasks where certain classes may dominate. This emphasizes HOCUS's ability to adapt and provide accurate transferability estimation even with multi-modal data.

5. Discussion and Conclusion

Discussion. In Sec. 4, we validate the effectiveness of HOCUS across a wide range of representative application scenarios. Despite its overall strong performance, several limitations remain. First, the linear correlation performance on the model zoo based on Vision Transformer (ViT) is not consistently strong. This behavior may stem from

[5] github.com/Tramac/awesome-semantic-segmentation-pytorch

the fundamental differences between convolutional inductive biases and self-attention mechanisms, which lead to distinct feature geometries in the embedding space. Second, on certain target datasets, HOCUS behaves close to a random predictor (*e.g.*, GTSRB in Table 1). In these cases, the fine-tuned performance gap among candidate models is extremely small, making reliable transferability estimation intrinsically difficult. Notably, this challenge is not unique to HOCUS, and nearly all existing transferability estimation methods exhibit similar limitations under such conditions. We consider these issues as open problems and plan to investigate them further in future work.

In addition, we discuss the explainability of HOCUS. Although HOCUS is not designed as a post-hoc explainable AI (XAI) method, it is inherently interpretable by construction. The proposed transferability score is explicitly decomposed into class separability and class uniformity, both of which are computed from well-defined feature-space statistics with clear geometric meanings. In particular, the class uniformity score provides an intuitive explanation of how evenly a pre-trained model represents different target classes, thereby offering insights into model bias that directly influence transferability. As future work, this interpretability can be further enhanced by integrating HOCUS with existing XAI techniques to provide more fine-grained explanations of model behavior.

Conclusion. This paper introduces a new perspective on transferability estimation by explicitly accounting for the class-wise bias of pre-trained models, a factor that has been largely overlooked by existing methods. By jointly considering class separability and class uniformity, HOCUS provides a more holistic assessment of transferability beyond conventional discrimination-focused metrics. In particular, our study highlights that strong class discrimination alone is insufficient for a reliable initial model, and that balanced representation across target classes plays a critical role.

HOCUS is simple, scalable, and model-agnostic, making it readily applicable to diverse model zoos and downstream tasks. Moreover, the proposed class uniformity score enhances existing transferability metrics by addressing their limitations in handling class bias. Extensive experimental results demonstrate that incorporating class uniformity leads to more robust and consistent model ranking across varied settings.

We hope our work offers valuable insights and advances the community's understanding of transferability in pre-trained models.

Acknowledgments

We would like to sincerely thank the anonymous reviewers for their constructive comments and valuable suggestions, which have greatly helped us improve the quality and clarity of this manuscript. We are also grateful to the editor for the careful handling of our submission and for providing insightful guidance throughout the review process. Their efforts and feedback are deeply appreciated.

References

[1] Y. Ding, B. Jiang, A. Yu, A. Zheng, J. Liang, Which model to transfer? a survey on transferability estimation, arXiv preprint arXiv:2402.15231 (2024).

[2] Y. Bao, Y. Li, S.-L. Huang, L. Zhang, L. Zheng, A. Zamir, L. Guibas, An information-theoretic approach to transferability in task transfer learning, in: Proc. ICIP, 2019, pp. 2309–2313.

[3] A. T. Tran, C. V. Nguyen, T. Hassner, Transferability and hardness of supervised classification tasks, in: Proc. ICCV, 2019, pp. 1395–1405.

[4] V. Papyan, X. Han, D. L. Donoho, Prevalence of neural collapse during the terminal phase of deep learning training, Proceedings of the National Academy of Sciences 117 (2020) 24652–24663.

[5] M. Pándy, A. Agostinelli, J. Uijlings, V. Ferrari, T. Mensink, Transferability estimation using bhattacharyya class separability, in: Proc. CVPR, 2022, pp. 9172–9182.

[6] Z. Wang, Y. Luo, L. Zheng, Z. Huang, M. Baktashmotlagh, How far pre-trained models are from neural collapse on the target dataset informs their transferability, in: Proc. ICCV, 2023, pp. 5549–5558.

[7] P. K. Khoba, Z. Wang, C. Arora, M. Baktashmotlagh, Feature space perturbation: A panacea to enhanced transferability estimation, in: Proc. WACV, 2025, pp. 1299–1308.

[8] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.

[9] X. Wang, Y. Zhao, C. Li, P. Ren, Probsap: A comprehensive and high-performance system for student academic performance prediction, Pattern Recognition 137 (2023) 109309.

[10] Y. Zhang, J. Hu, D. Wen, W. Deng, Unsupervised evaluation for out-of-distribution detection, Pattern Recognition (2024) 111212.

[11] X. Du, Z. Liu, Z. Feng, H. Deng, Datamap: Dataset transferability map for medical image classification, Pattern Recognition 146 (2024) 110044.

[12] C. Nguyen, T. Hassner, M. Seeger, C. Archambeau, Leep: A new measure to evaluate transferability of learned representations, in: Proc. ICML, 2020, pp. 7294–7305.

[13] A. Agostinelli, J. Uijlings, T. Mensink, V. Ferrari, Transferability metrics for selecting source model ensembles, in: Proc. CVPR, 2022, pp. 7936–7946.

[14] K. You, Y. Liu, J. Wang, M. Long, Logme: Practical assessment of pre-trained models for transfer learning, in: Proc. ICML, 2021, pp. 12133–12143.

[15] W. Shao, X. Zhao, Y. Ge, Z. Zhang, L. Yang, X. Wang, Y. Shan, P. Luo, Not all models are equal: Predicting model transferability in a self-challenging fisher space, in: Proc. ECCV, 2022, pp. 286–302.

[16] T. Zhang, Y. Shu, X. Chen, Y. Long, C. Guo, B. Yang, Assessing pre-trained models for transfer learning through distribution of spectral components, in: Proc. AAAI, volume 39, 2025, pp. 22560–22568.

[17] P. K. Khoba, Z. Wang, C. Arora, M. Baktashmotlagh, Peftdiff: Diffusion-guided transferability estimation for parameter-efficient fine-tuning, in: Proc. ICCV, 2025, pp. 1454–1463.

[18] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, S. Savarese, Taskonomy: Disentangling task transfer learning, in: Proc. CVPR, 2018, pp. 3712–3722.

[19] Y. Jiang, D. Krishnan, H. Mobahi, S. Bengio, Predicting the generalization gap in deep networks with margin distributions, arXiv preprint arXiv:1810.00113 (2018).

[20] R. Xie, H. Wei, L. Feng, Y. Cao, B. An, On the importance of feature separability in predicting out-of-distribution error, in: Proc. NeurIPS, volume 36, 2023, pp. 27783–27800.

[21] O. Zohar, S.-C. Huang, K.-C. Wang, S. Yeung, Lovm: Language-only vision model selection, in: Proc. NeurIPS Workshops, volume 36, 2023, pp. 33120–33132.

[22] Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, Q. Qu, A geometric analysis of neural collapse with unconstrained features, in: Proc. NeurIPS, 2021, pp. 29820–29834.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proc. CVPR, 2009, pp. 248–255.

[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. CVPR, 2016, pp. 770–778.

[25] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proc. CVPR, 2017, pp. 4700–4708.

[26] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proc. ICCV, 2019, pp. 1314–1324.

[27] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: Proc. ICML, 2019, pp. 6105–6114.

[28] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. ICLR, 2015.

[29] A. Krizhevsky, Learning multiple layers of features from tiny images, Master's thesis, University of Tront (2009).

[30] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proc. AISTATS, 2011, pp. 215–223.

[31] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: Proc. CVPR, 2012, pp. 3498–3505.

[32] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[33] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel, Detection of traffic signs in real-world images: The german traffic sign detection benchmark, in: Proc. IJCNN, 2013, pp. 1–8.

[34] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proc. CVPR, 2014, pp. 3606–3613.

[35] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: Proc. CVPR, 2010, pp. 3485–3492.

[36] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–mining discriminative components with random forests, in: Proc. ECCV, 2014, pp. 446–461.

[37] M.-E. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: Proc. CVPR, 2006, pp. 1447–1454.

[38] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: Proc. ICCV, 2013.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: Proc. ICML, 2021, pp. 8748–8763.

[40] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: Proc. NeurIPS Workshops, 2011.

[41] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, arXiv preprint arXiv:1306.5151 (2013).

[42] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: Proc. ICLR, 2018.

[43] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proc. ICCV, 2019, pp. 6023–6032.

[44] T. DeVries, G. W. Taylor, Improved regularization of convolutional neural networks with cutout, arXiv preprint arXiv:1708.04552 (2017).

[45] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, arXiv preprint arXiv:1612.02295 (2016).

[46] K. Banerjee, R. R. Gupta, K. Vyas, B. Mishra, et al., Exploring alternatives to softmax function, arXiv preprint arXiv:2011.11538 (2020).

[47] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proc. CVPR, 2017, pp. 2881–2890.

[48] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proc. CVPR, 2017, pp. 633–641.

[49] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge 2012 results, vol. 5 (2012), 2012.

[50] T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, D. Samaras, Large-scale training of shadow detectors with noisily-annotated shadow examples, in: Proc. ECCV, 2016, pp. 816–832.

[51] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Proc. ECCV, 2014, pp. 740–755.

[52] K. Gong, X. Liang, D. Zhang, X. Shen, L. Lin, Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing, in: Proc. CVPR, 2017, pp. 932–940.

[53] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: Proc. CVPR, 2012.

[54] G. J. Brostow, J. Fauqueur, R. Cipolla, Semantic object classes in video: A high-definition ground truth database, Pattern Recognition Letters 30 (2009) 88–97.

[55] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proc. CVPR, 2016, pp. 3213–3223.

Highlights

- We introduce a simple yet effective transferability estimation framework termed **H**arm**O**nizing **C**lass **U**niformity and **S**eparability (HOCUS), including a basic class separability score and an additional class uniformity score.
- We propose a novel class uniformity score (CU), formulated as the entropy of the class distribution overlapping matrix. CU is flexible and can be easily integrated as a plug-and-play score into existing methods.
- We conduct experiments on both image classification and semantic segmentation tasks. We also consider various model zoos involving multiple model architectures, multiple loss functions, and multi-source datasets. Experimental results demonstrate that HOCUS yields state-of-the-art results for transferability estimation.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: