Duality Diagram Similarity: a generic framework for initialization selection in task transfer learning

Kshitij Dwivedi$^{1,3[0000-0001-6442-7140]}$, Jiahui Huang$^{2[0000-0002-0389-1721]}$,
Radoslaw Martin Cichy$^{3[0000-0003-4190-6071]}$, and
Gemma Roig$^{1[0000-0002-6439-8076]}$

[1] Department of Computer Science, Goethe University Frankfurt, Germany
kshitijdwivedi93@gmail.com, roig@cs.uni-frankfurt.de
[2] ISTD, Singapore University of Technology and Design, Singapore
jiahui_huang@sutd.edu.sg
[3] Department of Education and Psychology, Free University Berlin, Germany
rmcichy@zedat.fu-berlin.de

Abstract. In this paper, we tackle an open research question in transfer learning, which is selecting a model initialization to achieve high performance on a new task, given several pre-trained models. We propose a new highly efficient and accurate approach based on duality diagram similarity (DDS) between deep neural networks (DNNs). DDS is a generic framework to represent and compare data of different feature dimensions. We validate our approach on the Taskonomy dataset by measuring the correspondence between actual transfer learning performance rankings on 17 taskonomy tasks and predicted rankings. Computing DDS based ranking for 17×17 transfers requires less than 2 minutes and shows a high correlation (0.86) with actual transfer learning rankings, outperforming state-of-the-art methods by a large margin (10%) on the Taskonomy benchmark. We also demonstrate the robustness of our model selection approach to a new task, namely Pascal VOC semantic segmentation. Additionally, we show that our method can be applied to select the best layer locations within a DNN for transfer learning on 2D, 3D and semantic tasks on NYUv2 and Pascal VOC datasets.

Keywords: Transfer Learning, Deep Neural Network Similarity, Duality Diagram Similarity, Representational Similarity Analysis

1 Introduction

Deep Neural Networks (DNNs) are state-of-the-art models to solve different visual tasks, *c.f.* [18,42]. Yet, when the number of training examples with labeled data is small, the models tend to overfit during training. To tackle this issue, a common approach is to use transfer learning by selecting a pre-trained network on a large-scale dataset and use it as initialization [28,19]. But how does one choose the model initialization that yields the highest accuracy performance when learning a new task?

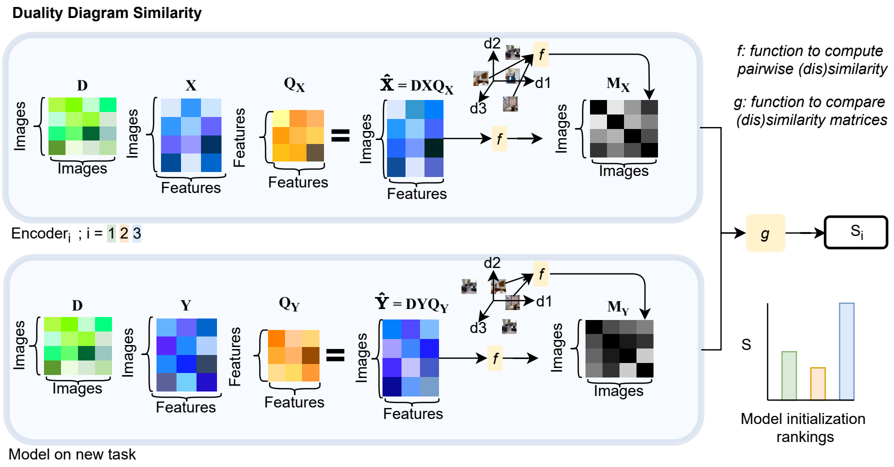

Duality Diagram Similarity
f: function to compute pairwise (dis)similarity
g: function to compare (dis)similarity matrices
D
X
Q_X
X̂ = DXQ_X
M_X
Images
Features
d1
d2
d3
f
g
S_i
Encoder_i ; i = 1 2 3
Y
Q_Y
Ŷ = DYQ_Y
M_Y
S
Model initialization rankings
Model on new task

Fig. 1. *Duality Diagram Similarity (DDS):* We apply DDS to compare features of a set of initialization options (encoders) with features of a new task to get model initialization rankings to select the encoder initialization for learning a new task. The task feature for an image is obtained by doing a feedforward pass through a model trained on that task. $\mathbf{D}$ is the matrix that weights the images, $\mathbf{X}$ ($\mathbf{Y}$) is the matrix that stores the features from the encoder for all images, $\mathbf{Q_X}$ ($\mathbf{Q_Y}$) is a matrix that stores relations between features dimensions, $\mathbf{M_X}$ ($\mathbf{M_Y}$) contains the pairwise (dis)similarity distances between images, and S_i is the score for the ranking.

Nowadays, there are a plethora of online available pre-trained models on different tasks. However, there are only a few methods [8,35] that automatically assist in selecting an initialization given a large set of options. Due to lack of a standard benchmark with standard evaluation metrics, comparing and building upon these methods is not trivial. Recently, Dwivedi and Roig [8] and Song *et al.* [35] used the transfer learning performance on the Taskonomy dataset [42] as groundtruth to develop methods for model selection. Both aforementioned methods for model selection are efficient compared to the bruteforce approach of obtaining transfer performance from all the models and selecting the best one. Yet, they used different metrics to evaluate against the groundtruth, and hence, they are not comparable in terms of accuracy. Although different, both of them used metrics that evaluate how many models in top-K ranked model initializations according to transfer learning performance were present in the top-K ranked models obtained using their method. We argue that such a metric doesn't provide a complete picture as it ignores the ranking within the top-K models as well as the ranking of models not in the top-K.

In this work, we first introduce a benchmark with a standard evaluation metric using Taskonomy [42] transfer learning dataset to compare different model initialization selection methods. We use Spearman's correlation between the rankings of different initialization options according to transfer learning performance and the rankings based on a model initialization selection method as our metric for comparison. We argue that our proposed benchmark will facilitate the comparison of existing and new works on model selection for transfer learning. We then introduce a duality diagram [9,12,6] based generic framework to compare DNN features which we refer to as duality diagram similarity (DDS). Duality diagram expresses the data taking into account the contribution of individual observations and individual feature dimensions, and the interdependence between observations as well as feature dimensions (see Fig. 1). Due to its generic nature, it can be shown that recently introduced similarity functions [8,17] for comparing DNN features are special cases of the general DDS framework.

We find that model initialization rankings using DDS show very high correlation (>0.84) with transfer learning rankings on Taskonomy tasks and outperform state-of-the-art methods [8,35] by a 10% margin. We also demonstrate the reliability of our method on a new dataset and task (PASCAL VOC semantic segmentation) in the experiments section.

Previous works [41,22] have shown the importance of selecting which layer in the network to transfer from. In this paper, we also explore if the proposed method could be used to interpret representations at different depths in a pre-trained model, and hence, it could be used to select from which layer the initialization of the model should be taken to maximize transfer learning performance. We first show that the representation at different blocks of pre-trained ResNet [11] model on ImageNet [7] varies from 2D in block1, to 3D in block 3 and semantic in block 4. These observations suggest that representation at different depths of the network is suitable for transferring to different tasks. Tranfer learning experiments using different blocks in a ResNet-50 trained on ImageNet as initialization for 2D, 3D, and semantic tasks on both, NYUv2 [23] and Pascal VOC [10] datasets, reveal that it is indeed the case.

2 Related Works

Our work relies on comparing DNN features to select pre-trained models as initialization for transfer learning. Here, we first briefly discuss related literature in transfer learning, and then, different methods to compare DNN features.

Transfer Learning. In transfer learning [25] the representations from a source tasks are re-used and adapted to a new target task. While transfer learning in general may refer to task transfer[28,19,40,42], or domain adaptation[30,37], in this work we focus specifically on task transfer learning. Razavian *et al.* [32] showed that features extracted from Overfeat [31] network trained on ImageNet [7] dataset can serve as a generic image representation to tackle a wide variety of recognition tasks. ImageNet pre-trained models also have been used to transfer to a diverse range of other vision related tasks [28,5,19,40]. Other works [17,14] have investigated why ImageNet trained models are good for transfer learning. In contrast, we are interested in improving the transfer performance by finding a better initialization that is more related to the target task.

Azizpour *et al.* [1] investigated different transferability factors. They empirically verified that the effectiveness of a factor is highly correlated with the distance between the source and target task distance obtained with a predefined categorical task grouping. Zamir *et al.* [42] showed in a fully computational manner that initialization matters in transfer learning. Based on transfer performance they obtained underlying task structure that showed clusters of 2D, 3D, and semantic tasks. They introduced the Taskonomy dataset [42], which provides pre-trained models on over 20 single image tasks with transfer learning performance on each of these tasks with every pre-trained model trained on other tasks as the initialization, and thus, providing groundtruth for a large number of transfers. Recent works [8,35] have used the Taskonomy transfer performance as groundtruth to evaluate methods of estimating task transferabilities. Those works use different evaluation metrics, which makes the comparison between those methods difficult. Following those works, we use Taskonomy transfer performance as a benchmark, and propose a unified evaluation framework to facilitate comparison between existing and future methods.

Yosinski *et al.* [41] explored transferability at different layers of a pre-trained network, and Zhuo *et al.* [44] showed the importance of focusing on convolutional layers of the model in domain adaptation. We also investigate if the similarity between DNN representations can be applied to both model and layer selection for transfer learning, which indeed is the case, as we show in the results section.

Similarity Measures for Transfer Learning Performance. Our approach is built under the assumption that the higher the similarity between representations is, the higher will be the transfer learning performance. Some previous works used similarity measures to understand the properties of DNNs. Raghu *et al.* [26] proposed affine transform invariant measure called Singular Vector Canonical Correlation Analysis (SVCCA) to compare two representations. They applied SVCCA to probe the learning dynamics of neural networks. Kornblith *et al.* [16] introduced centered kernel alignment (CKA) that shows high reliability in identifying correspondences between representations in networks trained using different initializations. However, in the above works, the relation between similarity measures and transfer learning was not explored.

Dwivedi and Roig [8] showed that Representational Similarity Analysis (RSA) can be used to compare DNN representations. They argued that using the model parameters from a model that has a similar representation to the new task's representation as initialization, should give higher transfer learning performance compared to an initialization from a model with a lower similarity score. Recently, Song *et al.* [35] used attribution maps [34,2,33] to compare two models and showed that it also reflects transfer learning performance. Our work goes beyond the aforementioned ones. Besides proposing an evaluation metric to set up a benchmark for comparison of these methods, we introduce a general framework using duality diagrams for similarity measures. We show that similarity measures, such as RSA and CKA, can be posed as a particular case in our gen-

Distances	Pearson's: $1 - \frac{(\mathbf{x}_i - \overline{\mathbf{x}_i}) \cdot (\mathbf{x}_j - \overline{\mathbf{x}_j})}{\lVert\mathbf{x}_i - \overline{\mathbf{x}_i}\rVert \cdot \lVert\mathbf{x}_j - \overline{\mathbf{x}_j}\rVert}$	Euclidean: $\sqrt{\mathbf{x}_i^T.\mathbf{x}_i + \mathbf{x}_j^T.\mathbf{x}_j - 2 * \mathbf{x}_i^T.\mathbf{x}_j}$	cosine: $1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\lVert\mathbf{x}_i\rVert \cdot \lVert\mathbf{x}_j\rVert}$
Kernels	linear: $\mathbf{x}_i^T \mathbf{x}_j$	Laplacian: $\exp(-\gamma_1 \lVert\mathbf{x}_i - \mathbf{x}_j\rVert_1)$	RBF: $\exp(-\gamma_2 \lVert\mathbf{x}_i - \mathbf{x}_j\rVert^2)$

Table 1. *Distance and Kernel functions used in DDS.* Notation: $\mathbf{x}_i \in \mathbb{R}^{d1}$ and $\mathbf{x}_j \in \mathbb{R}^{d1}$ refer to the features corresponding to i^{th} and j^{th} image (i^{th} and j^{th} row of feature matrix $\mathbf{X}$), respectively. Here, γ_1 and γ_2 refer to the bandwidth of Laplacian and RBF kernel.

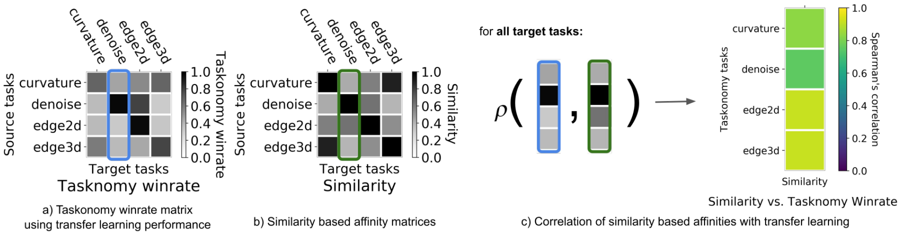
eral formulation. It also allows to use other more powerful similarities that are more highly correlated to transfer learning performance.

There is evidence in the deep learning literature, that normalization plays a crucial role. For instance, batch normalization allows training of deeper networks [15], efficient domain adaptation [20,3] and parameter sharing across multiple domains [27]. Instance normalization improves the generated image quality in fast stylization [38,13], and group normalization stabilizes small batch training [39]. In our DDS generic framework, it is straightforward to incorporate feature normalization. Thus, we further take into account the normalization of features before assessing the similarity between two DNN features and compare it to transfer learning performance.

3 Duality Diagram Similarity (DDS)

The term duality diagram was introduced by Escoufier [9] to derive a general formula of Principal Component Analysis that takes into account change of scale, variables, weighing of feature dimensions and elimination of dependence between samples. With similar motivation, we investigate the application of duality diagrams in comparing two DNNs. Let $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ refer to a matrix of features with dimensionality d_1 obtained from feedforwarding n images through a DNN. The duality diagram of matrix $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ is a triplet ($\mathbf{X}$,$\mathbf{Q}$,$\mathbf{D}$) consisting of a matrix $\mathbf{Q} \in \mathbb{R}^{d_1 \times d_1}$ that quantifies dependencies between the individual feature dimensions, and a matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ that assigns weights on the observations, *i.e.*, images in our case. Hence, a DNN representation for a set of n examples can be expressed by its duality diagram. By comparing duality diagrams of two DNNs we can obtain a similarity score. We denote the two DNN duality diagrams as ($\mathbf{X}$,$\mathbf{Q}_\mathbf{X}$,$\mathbf{D}$) and ($\mathbf{Y}$,$\mathbf{Q}_\mathbf{Y}$,$\mathbf{D}$), in which the subindices in the matrix $\mathbf{Q}$ denote that they are computed from the set of features and images in $\mathbf{X}$ and $\mathbf{Y}$.

To compare two duality diagrams, Robert and Escoufier [29] introduced the RV coefficient. The motivation behind RV coefficient was to map n observations of $\mathbf{X}$ in the d_1-dimensional space and $\mathbf{Y}$ in the d_2-dimensional space. Then, the similarity between $\mathbf{X}$ and $\mathbf{Y}$ can be assessed by comparing the pattern of obtained maps or, equivalently, by comparing the set of distances between all pairwise observations of both maps. To estimate the distances between pair-



a) Taskonomy winrate matrix using transfer learning performance

b) Similarity based affinity matrices

c) Correlation of similarity based affinities with transfer learning

Fig. 2. *Transfer learning vs. similarity measures.* We consider a) Taskonomy winrate matrix, b) an affinity matrix obtained by measuring similarity between DNNs trained on different tasks. c) the Spearman's correlation (denoted by ρ) between the columns of two matrices. The resulting vector shows the correlation of the similarity based rankings with transfer learning performance based rankings for 4 Taskonomy tasks. Here we illustrate the results using DDS ($f = Laplacian$), and the procedure remains the same using any similarity measure.

wise observation, Robert and Escoufier [29] used dot product and compared two (dis)similarity matrices using the cosine distance.

In a nutshell, to compare two sets of DNN features $\mathbf{X}$ and $\mathbf{Y}$, we require three steps (Fig. 1): first, transforming the data using $\mathbf{Q_X}$ and $\mathbf{D}$ to $\hat{\mathbf{X}}$, using $\hat{\mathbf{X}} = \mathbf{DXQ_X}$, and $\hat{\mathbf{Y}}$ with $\hat{\mathbf{Y}} = \mathbf{DYQ_Y}$. Second, using a function, which we denote as f, to measure (dis)similarity between each pair of data points to generate pairwise distance maps. Let $\mathbf{M_X}$ be the matrix that stores the (dis)similarity between pairwise distance maps for $\hat{\mathbf{X}}$, also referred to as representational (dis)similarity matrices. It is computed as $\mathbf{M_X}(i,j) = f(\hat{\mathbf{X}}(i,:), \hat{\mathbf{X}}(j,:))$, in which i and j denote the indices of the matrices. Analogously, $\mathbf{M_Y}$ is the matrix that stores the (dis)similarity between pairwise distance maps of $\hat{\mathbf{Y}}$. Third, a function g to compare $\mathbf{M_X}$ and $\mathbf{M_Y}$ to obtain a final similarity score, denoted as S, and computed as $S = g(\mathbf{M_X}, \mathbf{M_Y})$ is applied. The above formulation using duality diagrams provides a general formulation that allows us to investigate empirically which combination of $\mathbf{Q}$,$\mathbf{D}$, f and g is suitable for a given application, which in our case is estimating transferability rankings to select the best model (or layer in a model) to transfer given a new dataset and/or task.

Interestingly, using the above DDS framework, we can easily show that recently used similarity measures, e.g., CKA and RSA, can be formulated as special cases of DDS. For RSA [8], $\mathbf{Q}$ is an identity matrix, $\mathbf{I} \in \mathbb{R}^{d_1 \times d_1}$, and $\mathbf{D}$ is a centering matrix, *i.e.*, $\mathbf{C} = \mathbf{I_n} - \frac{1}{n}\mathbf{1_n}$. f is Pearson's distance and g is Spearman's correlation between lower/upper triangular part of $\mathbf{M_X}$ and $\mathbf{M_Y}$. For CKA [17], $\mathbf{Q}$ and $\mathbf{D}$ are identity matrices $\mathbf{I} \in \mathbb{R}^{d_1 \times d_1}$ and $\mathbf{I} \in \mathbb{R}^{n \times n}$ respectively, f used is linear or RBF kernel and g is cosine distance between unbiased centered (dis)similarity matrices. In the supplementary section S1, we derive RSA and CKA as particular cases of the DDS framework.

In this work, we focus on exploring different instantiations of $\mathbf{Q}$, $\mathbf{D}$, f and g from our DDS framework that are most suitable for estimating transfer learning performance. We consider different formulations of $\mathbf{Q}$ and $\mathbf{D}$, resulting in z-scoring, batch normalization, instance normalization, layer normalization and group normalization (details in Supplementary S2). For function f we explore cosine, Euclidean, and Pearson's distance, as well as kernel based similarities, namely linear, RBF, and Laplacian. Mathematical equations for all functions are in Table 1. For function g, we consider Pearson's correlation to compare (dis)similarity matrices with and without unbiased centering [36,17].

4 Our Approach

4.1 Which DDS combination (Q, D, f ,g) best predicts transferability?

After having defined the general formulation for using similarity measures for transfer learning, we can instantiate each of the parameters ($\mathbf{Q}$, $\mathbf{D}$, f and g) to obtain different similarity measures. To evaluate which combination of $\mathbf{Q}$, $\mathbf{D}$, f and g best predicts transferability and compare it to state-of-the-art methods, we consider transfer learning performance based winrate matrix (Fig. 2a) and affinity matrix proposed in Taskonomy dataset [42], as a transferability benchmark. The affinity matrix is calculated by using actual transfer learning performance on the target task given multiple source models pre-trained on different tasks. The winrate matrix is calculated using a pairwise competition between all feasible sources for transferring to a target task. Both these matrices represent transfer learning performance obtained by bruteforce, and hence, can be considered as an upper bound for benchmarking transferability. We use the Taskonomy dataset as a benchmark as it consists of pre-trained models on over 20 single image tasks with transfer learning performance on each of these tasks with every pre-trained model trained on other tasks as the initialization, thus, providing groundtruth for a large number of task transfers.

We use DDS to quantify the similarity between two models trained on different Taskonomy tasks and use that value to compute the DDS based affinity matrix (Fig. 2b). A column vector corresponding to a specific task in the Taskonomy affinity matrix shows the transfer learning performance on the target task when different source tasks were used for initialization. To evaluate how well a DDS based affinity matrix represents transferability, we calculate the Spearman's correlation between columns of the Taskonomy winrate/affinity matrix and DDS based affinity matrix. Using the rank-based Spearman's correlation for comparison between two rankings allows comparing the source tasks ranking on the basis of transfer learning performance with DDS based ranking. The resulting vector (Fig. 2c) represents the per task correlation of DDS with transferability.

We further evaluate if the best combination(s) we obtained from the above proposed evaluation benchmark using Taskonomy are robust to a new dataset and task. For this, we consider a new task, Pascal VOC semantic segmentation, following [8]. For the benchmark, we use the transfer learning performance on Pascal VOC semantic segmentation task given all Taskonomy models as sources.

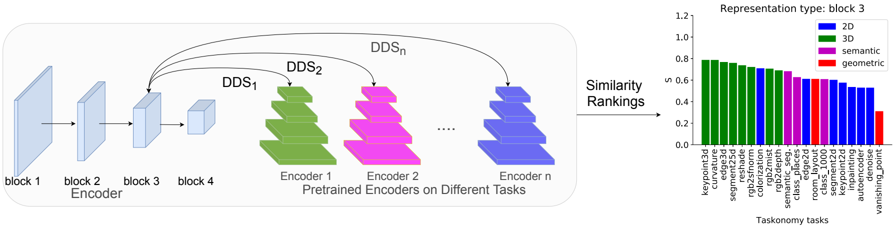


Fig. 3. *DNN Layer Selection.* Given a pre-trained encoder and a set of pre-trained models trained on diverse tasks, we can assess the representation type at different depth of the network by comparing the similarity between features at a given depth and pre-trained models.

We also investigate if the images selected to compute DDS have any effect on Spearman's correlation with transfer learning. For this purpose, we select images from NYUv2, Taskonomy, and Pascal VOC dataset and evaluate the proposed methods on both, Taskonomy and Pascal VOC benchmark. We further compute the variance performing bootstrap by randomly sampling 200 images from the same dataset 100 times to compute similarity. The bootstrap sampling generates a bootstrap distribution of correlation between transfer performance and similarity measures, which allows measuring the variance in Spearman's correlation with transfer performance when selecting different images from the same dataset.

4.2 Does DDS find best layer representation within a model to transfer from?

In previous works [42,8,35], a major focus was to select a model to initialize from. However, once the model is selected as an encoder for initialization, the new layers of decoder usually branch out from the last layer of the pre-trained encoder. Such an approach is based on the *a priori* assumption that, for any new task, the output from the last layer of the pre-trained encoder is the best representation for transfer learning. We argue that this is task-type dependent. For instance, it has been shown that earlier layers of DNNs trained on ImageNet object recognition learn low-level visual features while deeper layers learn high-level categorical features [24]. Therefore, one would expect for low-level visual task, the representation in earlier layers of DNN might be better for transfer learning. Based on this intuition, we investigate if layers at different depths of the network are better suited to transfer to different types of tasks. We compute DDS of a pre-trained model at different depths with Taskonomy models to assess representation types at different depths (Fig. 3). To validate it, we select 3 task types (2D, 3D, and semantic) from NYUv2 and Pascal VOC dataset and perform transfer learning by attaching the decoder to different encoder layers.

5 Experimental Setup

We implemented the DDS general framework in python[4], in which new parameters and functions (**Q**, **D**, f, g) can be incorporated in the future. Below, we first provide details of datasets and models used for comparing the DDS combinations for model selection. Then, we describe the datasets and models used for layer selection from a pre-trained encoder.

5.1 Dataset and models for Model Selection

Datasets. To compare different DDS combinations against the Taskonomy affinity and winrate matrix, we randomly select 200 images from the Taskonomy dataset. We use 200 images based on an analysis that shows that the correlation of DDS with transfer learning performance saturates at around 200 images (see Supplementary S3). To perform the bootstrap based comparison on a new semantic segmentation task on the Pascal VOC dataset, we randomly select 5000 images from Taskonomy, 5000 images from Pascal VOC, and all (1449) images from NYUv2 dataset.

Models. We use the selected 200 images to generate features from the last layer of the encoder of 17 models trained on 17 different tasks on the Taskonomy dataset. The Taskonomy models have an encoder/decoder architecture. The encoder architecture for all the tasks is a fully convolutional Resnet-50 without pooling to preserve high-resolution feature maps. The decoder architecture varies depending on the task. The models were trained on different tasks independently using the same input images but different labels corresponding to different tasks. For comparing two models, we use the features of the last layer of the encoder following [42,8]. The Pascal VOC semantic segmentation model that we use also has the same encoder architecture as Taskonomy models, and the decoder is based on the spatial pyramid pooling module, which is suitable for semantic segmentation tasks [5]. For comparison with the Pascal VOC model, we use the features of the last layer of the encoder of 17 Taskonomy models and the one Pascal VOC semantic segmentation model trained from scratch. We also report comparison with a small Pascal VOC model from [8] in Supplementary S4 to show that model selection can be performed even using small models.

5.2 Dataset and models for layer selection

Datasets. To validate whether the proposed layer selection using similarity measures reflects transferability, we perform training on different datasets and tasks by branching the decoders from different layers of the encoder. Specifically, we evaluate on 3 tasks (Edge Detection, Surface Normal Prediction and Semantics Segmentation) on Pascal VOC [10] dataset, and 3 tasks (Edge Detection, Depth Prediction and Semantic Segmentation) on NYUv2 [23] dataset. Following Zamir *et al.* [42], we use Canny Edge Detector [4] to generate groundtruth edge maps while other labels were downloaded from Maninis *et al.* [21].

[4] Code available at https://github.com/cvai-repo/duality-diagram-similarity

f	kernels			distances		
Q,**D**	linear	Laplacian	RBF	Pearson	euclidean	cosine
Identity	0.632	0.815	0. 800	0.823	0.688	0.742
Z-score	0.842	**0.860**	**0.841**	0.856	**0.850**	**0.864**
Batch norm	0.729	0.852	0.840	**0.857**	0.807	0.850
Instance norm	**0.849**	0.835	0.838	0.850	0.847	0.850
Layer norm	0.823	0.806	0.786	0.823	0.813	0.823
Group norm	0.829	0.813	0.790	0.829	0.814	0.829

Table 2. *Finding best DDS combination (**Q**, **D**, f ,g).* We report the results of comparison with transferability for differents sets of **Q**, **D** and f. Top 3 scores are shown in green, blue, brown, respectively. Best **Q**, **D** for each f is shown in bold.

Models. We describe the models' encoder and decoder.
Encoder: We use a ResNet-50 [11] pre-trained on ImageNet [7] as our encoder, which has four blocks, each of the block consist of several convolution layers with skip connections, followed by a pooling layer. The branching locations that we explore are after each of the four pooling layers. We also consider Resnet-50 pre-trained on Places [43] using the same experimental set-up, and report the results in Supplementary S5.
Decoder: Following the success of DeepLabV3 [5] model, we use their decoder architecture in all our experiments. Since the output channels of the ResNet-50 encoder varies at different branching locations, we stack the output feature maps to keep the number of parameters in the downstream constant. More specifically, the encoder outputs 256, 512, 1024, 2048 channels for location 1, 2, 3 and 4 respectively, we stack the output of early branchings multiple times (8× for location 1, 4× for location 2 and 2× for location 3) to achieve a constant 2048 output channels to input to the decoder.

Training. ImageNet [7] pre-trained encoder is fine-tuned for the specific tasks, while the decoder is trained from scratch. In all the performed experiments, we use synchronized SGD with momentum of 0.9 and weight decay of 1e-4. The initial learning rate was set to 0.001 and updated with the "poly" learning rate policy [5]. The total number of epochs for the training was set to 60 and 200, for Pascal VOC [10] and NYUv2 [23], respectively as in Maninis *et al.* [21].

6 Results

In this section, we first report the comparison results of different similarity measures. After selecting the best similarity measure we apply it for identifying the representation type at different depth of the pre-trained encoder. Finally, we validate if the branching selection suggested using similarity measures gives the best transfer performance, by training models with different branching locations on NYUv2 and Pascal VOC datasets.

Method	Affinity	Winrate	Total time(s)
Taskonomy Winrate[42]	0.988	1	1.6×10^7
Taskonomy affinity[42]	1	0.988	1.6×10^7
saliency[35]	0.605	0.600	3.2×10^3
DeepLIFT[35]	0.681	0.682	3.3×10^3
ϵ-LRP[35]	0.682	0.682	5.6×10^3
RSA[8]	0.777	0.767	78.2
DDS ($f = cosine$)	0.862	0.864	84.14
DDS ($f = Laplacian$)	0.860	0.860	103.36

Table 3. *Correlation of DDS based affinity matrices with Taskonomy affinity and winrate matrix, averaged for 17 Taskonomy tasks, and comparison to state-of-the-art.* Top 2 scores are shown in green, and blue respectively. For this experiment, **Q** and **D** are selected to perform z-scoring, in all DDS tested frameworks.

6.1 Finding best DDS combination (Q, D, f ,g) for transferability

We perform a thorough analysis to investigate which combinations of (**Q**, **D**, f, and g) of the DDS lead to higher correlation with transferability rankings. We focus on how to assign weights to different feature dimensions using **Q**, **D** and distance functions f to compute the pairwise similarity between observations. In Table 2, we report results on the correlation with transferability rankings showing the effect of applying combination of **Q** and **D** instantiated as identity, z-score, batch/instance/group/layer normalization, and using different distance/kernel function as f. For g we use Pearson's correlation on unbiased centered dissimilarity matrices because it consistently showed a higher correlation with transfer learning performance (Supplementary Section S6). We observed a similar trend in results using Spearman's correlation for g (Supplementary Section S7).

In Table 2 we report the mean correlation of all the columns of the Taskonomy winrate matrix with the corresponding columns of a DDS based affinity matrix, which serve as the measure for computing how each of the similarity measures best predicts the transferability performance for each model. We first observe the results when **Q** and **D** are identity matrices. Laplacian and RBF kernels outperform linear kernel. For distance functions, Pearson outperforms euclidean and cosine. A possible reason for the better performance of Pearson's could be due to its invariance to translation and scale.

We next observe the effect of normalization using appropriate **Q** and **D**. We observe that the correlation with transferability rankings improves for all distance and kernel functions especially for low-performance distance and kernel functions. The gain in improvement is highest using z-scoring in most of the cases. A possible reason for overall performance improvement is that applying z-scoring reduces the bias in distance computation due to feature dimensions having high magnitude but low variance. Hence, for our next experiments, we choose z-scoring and select the top performing f: Laplacian and cosine.

Method	Taskonomy	Pascal VOC	NYUv2
DDS ($f = cosine$)	0.525 ±0.057	0.722 ±0.049	0.518 ±0.034
DDS ($f = Laplacian$)	0.5779 ±0.050	0.765 ±0.038	0.521 ±0.029

Table 4. *DDS correlation with transfer learning for Pascal VOC Semantic Segmentation.* Here each row represents a particular distance/kernel function as f, and each column represents a dataset. The values in the table are bootstrap mean correlation and standard deviation of a particular similarity measure computed using the image from a particular dataset. Top score is shown in green.

6.2 Comparison with state-of-the-art on Taskonomy

We first compare the DDS based affinity matrices on the Taskonomy transferability benchmark. To quantify in terms of mean correlation across all the tasks, we report mean correlation with Taskonomy affinity and winrate matrix in Table 3. In Table 3 (also Supplementary Section S8), we observe that all the proposed DDS based methods outperform the state-of-the-art methods [35,8] by a large margin. DDS ($f = cosine$) improves [8] and [35] by 10.9% (12.6%) and 26.3% (26.6%) on affinity (winrate), respectively. We report the correlation of different DDS based rankings with the rankings based on winrate and task affinities for 17 Taskonomy tasks in Supplementary S9 and find that proposed DDS based methods outperform state-of-the-art methods for almost all the tasks. We also report comparison using PR curve following [35] in Supplementary S10.

To compare the efficiency of different methods with respect to bruteforce approach, we report the computational budget required for different methods. A single forward pass of Taskonomy models on Tesla V100 GPU takes 0.022 seconds. Thus, for 17 tasks and 200 feedforward passes for each task, the total time for feedforward pass is 74.8 sec. Hence, the DDS based methods are several orders of magnitude faster than bruteforce approach, used in the Taskonomy approach [42], that requires several GPU hours to perform transfer learning on all the models. The number reported in Table 3 for Taskonomy was calculated by taking the fraction of the total transfer time (47,886 hours for 3000 transfers) for 17^2 transfers used for comparison in this work. Further, the time for obtaining DDS based rankings takes only a few seconds on CPU and is an order of magnitude faster than attribution maps based methods.

6.3 Evaluating robustness on a new task and dataset

In the evaluation benchmark that we proposed, which was used in the above reported experiments, we considered models that were trained using images from the Taskonomy dataset, and the images used to compute the DDS were also from the same dataset. To evaluate the robustness of DDS against a new task and images used to compute DDS, we consider a new task, namely Pascal VOC semantic segmentation, and use images from different datasets to compute DDS. To evaluate effect of selecting different images within the same dataset, we perform bootstrap to estimate the variance in correlation with transferability.

In Table 4, we report the bootstrap mean and standard deviation of correlation of different similarity measures with transfer learning performance on the Pascal VOC semantic segmentation task. We observe that the similarity measures show a high correlation (>0.70 for f = *cosine* and >0.75 for f = *Laplacian*) when using images from Pascal VOC, but low correlation when using images from another dataset (Taskonomy and NYUv2). We also observed a similar trend in Taskonomy benchmark (Supplementary Section S11). Thus, the similarity measure is effective when using images from the same distribution as the training images for the model of the new task. We believe that using images from the same data distribution in DDS as the ones used to train the model on the new task for selecting the best initialization is important because the model in the new task is trained using data sampled from this distribution. Since high correlation of DDS (f = *Laplacian*) with transferability is obtained in all the investigated scenarios using the images from the dataset of the new task that we want to transfer to, we argue that this is the most suitable choice for estimating transferability as compared to other similarity measures and set-ups.

6.4 Finding representation type at different depth of a model

In the previous experiments, we demonstrated DDS ability to select models for transfer learning to a new task, given a set of source models. Here, we use DDS to interpret the representation type at different depths of the model, which would allow us to select which model layer to transfer from for a given type of task. For this purpose, we generate the features of the last layer of the encoder of 20 Taskonomy models to get the representation of each task type. We then compute the DDS (f = *Laplacian*) of Taskonomy features with each output block of the pre-trained ImageNet model. We use images from the same data distribution (Taskonomy) as used in the trained models to reveal the correlation with each layer and task type, as suggested in the previous experiment.

As shown in Fig. 4, we observe that the representation of block 1 is more similar to 2D models, block 3 is more similar to 3D and block 4 to semantic models. These results suggest that the representation of block 1 is better suited to transfer for 2D tasks, block 3 for 3D tasks and block 4 for semantic tasks. There is no clear preference for block 2. We observe a similar pattern with the pre-trained Places model (see Supplementary S5).

6.5 Does DDS predict best branching location from an encoder?

Here we report the results of transfer learning performances of 4 tasks: surface normal prediction on Pascal VOC [10], Depth Prediction on NYU Depth V2 [23], and edge detection and semantic segmentation on both datasets. These 4 tasks cover the 3 task clusters we observed in the previous section. The results are shown on Table 5. We report the qualitative comparison of different block outputs in Supplementary S5. We observe that branching out from block 3 gives the best performance on depth and surface normal, branching out from block 1 provides the best result on edge detection, and branching out from block 4 is

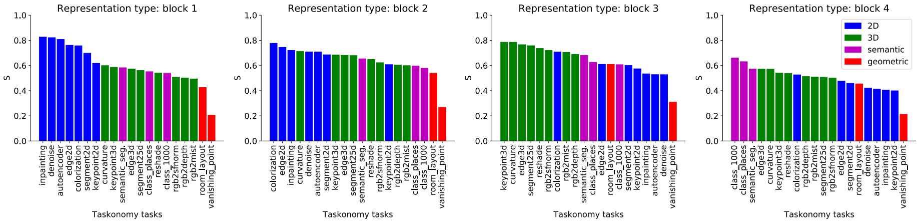


Fig. 4. *Block selection using DDS* on pre-trained encoder on Imagenet, and with DNNs trained on Taskonomy dataset on different tasks.

Task / Block	Pascal VOC			NYUv2		
	Edge (MAE)	Normals (mDEG_DIFF)	Semantic (mIOU)	Edge (MAE)	Depth (log RMSE)	Semantic (mIOU)
1	**0.658**	18.09	0.257	**0.823**	0.322	0.124
2	0.686	15.59	0.392	0.857	0.290	0.165
3	0.918	**14.39**	0.627	1.297	**0.207**	0.219
4	0.900	15.11	**0.670**	1.283	0.208	**0.285**

Table 5. *Transfer learning performance of branching ImageNet pre-trained encoder on different tasks on Pascal VOC and NYUv2.* Results show that branching out from block 1, 3, 4 of the encoder have better performances on edge, normals (depth) and semantic tasks, respectively. This is consistent with the diagram similarity in Fig. 4.

best for semantic segmentation. The transfer learning results are consistent with the similarity results in Fig. 4, which suggests that DDS (f = *Laplacian*) is a robust method for encoder block selection for different tasks.

7 Conclusion

In this work, we investigated duality diagram similarity as a general framework to select model initialization for transfer learning. We found that after taking into account the weighing of feature dimension, DDS (for all distance functions) show a high correlation (>0.84) with transfer learning performance. We demonstrated on Taskonomy models that the DDS (f = *Laplacian*,f = *cosine*) shows 10% improvement in correlation with transfer learning performance over state-of-the-art methods. DDS (f = *Laplacian*) is highly efficient and robust to novel tasks to create a duality diagram. We further show the DDS (f = *Laplacian*) effectiveness in layer selection within a model to transfer from.

Acknowledgments. G.R. thanks the support of the Alfons and Gertrud Kassel Foundation. R.M.C. is supported by DFG grants (CI241/1-1, CI241/3-1) and the ERC Starting Grant (ERC-2018-StG 803370).

References

1. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. IEEE transactions on pattern analysis and machine intelligence **38**(9), 1790–1802 (2015)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
3. Balaji, Y., Chellappa, R., Feizi, S.: Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 6499–6507 (2019)
4. Canny, J.: A computational approach to edge detection. In: Readings in computer vision, pp. 184–203. Elsevier (1987)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. ArXiv **abs/1706.05587** (2017)
6. De la Cruz, O., Holmes, S.: The duality diagram in data analysis: examples of modern applications. The annals of applied statistics **5**(4), 2266 (2011)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition pp. 248–255 (2009)
8. Dwivedi, K., Roig, G.: Representation similarity analysis for efficient task taxonomy & transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12387–12396 (2019)
9. Escoufier, Y.: The duality diagram: a means for better practical applications. In: Develoments in Numerical Ecology, pp. 139–156. Springer (1987)
10. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vision **88**(2), 303–338 (Jun 2010). https://doi.org/10.1007/s11263-009-0275-4, `http://dx.doi.org/10.1007/s11263-009-0275-4`
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2016)
12. Holmes, S., et al.: Multivariate data analysis: the french way. In: Probability and statistics: Essays in honor of David A. Freedman, pp. 219–233. Institute of Mathematical Statistics (2008)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
14. Huh, M., Agrawal, P., Efros, A.A.: What makes imagenet good for transfer learning? arXiv preprint arXiv:1608.08614 (2016)
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
16. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning. pp. 3519–3529 (2019)
17. Kornblith, S., Shlens, J., Le, Q.V.: Do better imagenet models transfer better? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2661–2671 (2019)

18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**, 84–90 (2012)
19. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. 2016 Fourth International Conference on 3D Vision (3DV) pp. 239–248 (2016)
20. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. ArXiv **abs/1603.04779** (2016)
21. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1851–1860 (2019)
22. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3994–4003 (2016)
23. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
24. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill (2017). https://doi.org/10.23915/distill.00007, https://distill.pub/2017/feature-visualization
25. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
26. Raghu, M., Gilmer, J., Yosinski, J., Sohl-Dickstein, J.: Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In: Advances in Neural Information Processing Systems. pp. 6076–6085 (2017)
27. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8119–8127 (2018)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
29. Robert, P., Escoufier, Y.: A unifying tool for linear multivariate statistical methods: the rv-coefficient. Journal of the Royal Statistical Society: Series C (Applied Statistics) **25**(3), 257–265 (1976)
30. Rozantsev, A., Salzmann, M., Fua, P.: Beyond sharing weights for deep domain adaptation. IEEE transactions on pattern analysis and machine intelligence **41**(4), 801–814 (2018)
31. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
32. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: 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)
33. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 (2016)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2015)
35. Song, J., Chen, Y., Wang, X., Shen, C., Song, M.: Deep model transferability from attribution maps. In: Advances in Neural Information Processing Systems (2019)
36. Székely, G.J., Rizzo, M.L., et al.: Partial distance correlation with methods for dissimilarities. The Annals of Statistics **42**(6), 2382–2412 (2014)

37. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167–7176 (2017)
38. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
39. Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
40. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1395–1403 (2015)
41. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014)
42. Zamir, A.R., Sax, A., Shen, W.: Taskonomy: Disentangling task transfer learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
43. Zhou, B., Khosla, A., Lapedriza, À., Torralba, A., Oliva, A.: Places: An image database for deep scene understanding. CoRR **abs/1610.02055** (2017)
44. Zhuo, J., Wang, S., Zhang, W., Huang, Q.: Deep unsupervised convolutional domain adaptation. In: MM '17 (2017)