A Statistical checks of the deconfounding assumption in Eq.4

A.1 Linearity assumption of the input similarity structure

In Eq.4, we assume that the representation similarity structures **linearly** depend on the input similarity structure. To validate if the linear assumption is sufficient, we check if adding higher-order polynomial terms of input similarity to the regression model (Eq.4) can help explain the representation similarity structure. We show the effect of adding higher-order polynomial terms on a pretrained ResNet-18 (contains 20 layers in total) with CIFAR-10 inputs in Figure 6. We observe that neither the R^2 nor the Bayesian information criterion (BIC) [47], approximating the model evidence, changes much when adding higher-order terms. Although R^2 can be marginally improved by increasing the order in the deeper layers (e.g., layer 20), we only consider the linear model in this paper for simplicity.

A.2 Independence assumption of the noise term across the dataset

In Eq.4, we consider the collection of distance between each pair of examples is the dataset of the linear regression. A potential issue is that the noise term $\boldsymbol{\epsilon}_f^m$ can be correlated for different pairs, while we assume independent noise when fitting model Eq.4. Because the noise $\epsilon_{f,ij}^m$ of the distance between example i and j and $\epsilon_{f,ik}^m$ of the distance between example i and k contain the same information about example i, which might induce correlation between $\epsilon_{f,ij}^m$ and $\epsilon_{f,ik}^m$. Checking noise correlation is important to justify if the model solution Eq.5 is misspecified.

For each i, we apply the Durbin-Watson (DW) test on $\{\epsilon_{f,ij}^m|\forall j\}$, and average the test statistics over the index i. DW is always in $[0, 4]$, and if there is no evidence of noise correlation, the test statistics equals to 2. Otherwise, the closer to 0 the statistic, the more evidence for positive correlation, while the closer to 4 means a negative correlation. We show the histogram of DW statistics as well as the averaged DW for each layer in Figure 6, where no serious noise correlation is observed.

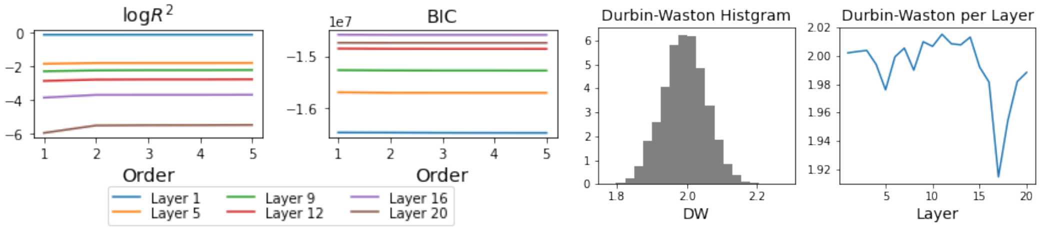

Figure 6: ***Left*: Log R^2 and BIC of regression models with different orders of input similarity**. We empirically observe that correcting for the confounder with a linear model is sufficient, especially for shallow layers (demonstrated with ResNet-18 on CIFAR-10). ***Right*: Durbin-Watson histogram and averaged DW statistics for each layer.**

B Proofs of propositions

B.1 Proof of Proposition 3.1

If the similarity measure $k(\cdot,\cdot)$ is invariant to orthogonal transformation, the representational similarity matrices are invariant to orthogonal transformation too. Thus, for any two full-rank orthonormal matrices U and V, such that $U^TU = I$ and $V^TV = I$, we have:

$$K(X_{f_1}^{m_1}U, X_{f_1}^{m_1}U) = K(X_{f_1}^{m_1}, X_{f_1}^{m_1});\;\; K(X_{f_2}^{m_2}V, X_{f_2}^{m_2}V) = K(X_{f_2}^{m_2}, X_{f_2}^{m_2}), \tag{14}$$

and the deconfounded RSMs are also invariant to orthogonal transformation:

$$\begin{aligned} dK(X_{f_1}^{m_1}U, X_{f_1}^{m_1}U) &= K(X_{f_1}^{m_1}U, X_{f_1}^{m_1}U) - \hat{\alpha}_{f_1}^{m_1}K^0 = K(X_{f_1}^{m_1}, X_{f_1}^{m_1}) - \hat{\alpha}_{f_1}^{m_1}K^0 = dK(X_{f_1}^{m_1}, X_{f_1}^{m_1}); \\ dK(X_{f_2}^{m_2}V, X_{f_2}^{m_2}V) &= K(X_{f_2}^{m_2}U, X_{f_2}^{m_2}U) - \hat{\alpha}_{f_2}^{m_2}K^0 = K(X_{f_2}^{m_2}, X_{f_2}^{m_2}) - \hat{\alpha}_{f_2}^{m_2}K^0 = dK(X_{f_2}^{m_2}, X_{f_2}^{m_2}). \end{aligned} \tag{15}$$

Therefore the deconfounded similarity is invariant to orthogonal transformation:

$$s(dK(X_{f_1}^{m_1}U, X_{f_1}^{m_1}U), dK(X_{f_2}^{m_2}V, X_{f_2}^{m_2}V)) = s(dK(X_{f_1}^{m_1}, X_{f_1}^{m_1}), dK(X_{f_2}^{m_2}, X_{f_2}^{m_2})), \tag{16}$$

which completes the proof.

Note that the PSD approximation will not affect the orthogonal invariance property, because deconfounded RSMs are invariant to orthogonal transformations.

B.2 Proof of Proposition 3.2

We use dRSA as an example to show the proof. For any $\gamma, \theta \in \mathbb{R}$, we have:

$$K(\gamma X^{m_1}_{f_1}, \gamma X^{m_1}_{f_1}) = \gamma^2 K(X^{m_1}_{f_1}, X^{m_1}_{f_1}); \quad K(\theta X^{m_2}_{f_2}, \theta X^{m_2}_{f_2}) = \theta^2 K(X^{m_2}_{f_2}, X^{m_2}_{f_2}), \tag{17}$$

because of using the Euclidean distance. Moreover, the deconfounded RSMs are scaled with λ^2 and θ^2 as well, because the regression coefficient, α in Eq.5, is scaled with the same value. Therefore, the rank correlation between two scaled deconfounded RSMs does not change because the rank is invariant to scaling all objects.

The PSD approximation will not affect the isotropic scaling invariance property too, because the PSD approximation matrix will be scaled at the same level.

C Training details of NNs

C.1 ResNet training on CIFAR-10

We trained 50 ResNet-18 models without any pretrained model from different initializations on the CIFAR-10 training set. We use 200 epochs with batch size 128. We train models with SGD: learning rate 0.1, momentum 0.9, and weight decay 5e-4. We also use cosine annealing learning rate with T$_{max} = 200$. The averaged accuracy of trained models on CIFAR-10 test set is 0.89, with standard deviation 0.3%. Model training takes around 10 hours.

C.2 ResNet training on DomainNet

We finetune Resnet-50 models which had been pretrained on the Imagenet dataset. Separate models are trained on each domain of the DomainNet dataset, namely Real, Clipart, Sketch and Quickdraw. The DomainNet task is to classify each image among 345 different classes. For each domain, we repeat the training for 10 random restarts. To keep the training uniform, we sample 5000 images from each domain and train the Resnet model for 2000 iterations with a batch size of 32. All the input images are resized to 224×224 pixels, and we perform no other form of data augmentation. We use the AdamW optimizer [48] with a base learning rate of $1e-3$, which was varied using a cosine annealing scheduler with a warm-up of 600 steps. The fine-tuning takes around 8 hours in total for all random seeds.

Figure 7 *Left* shows the F1 scores of the finetuned model for each domain. As expected, the score for the Real domain is the highest since it is the most similar to Imagenet. The higher score on the Quickdraw domain is possibly due to the simplicity of the Quickdraw input distribution (since it consists of doodles) as compared to the other domains.

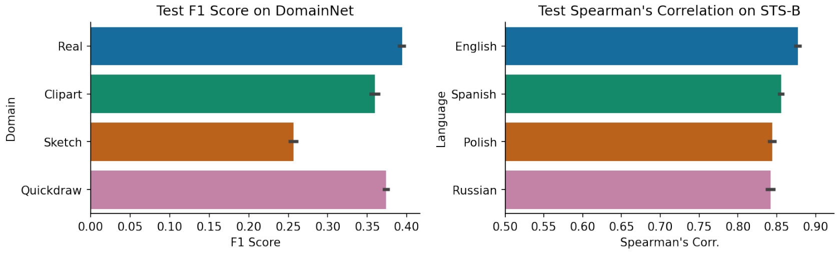

Figure 7: *Left*: Test F1 scores for each of the domains from DomainNet for Resnet-50 models. We can see that the Real domain, which is the most similar to Imagenet achieves the highest F1 score. *Right*: Test Spearman's correlation for the XLM-RoBERTa model when finetuned on each language.

C.3 XLM-RoBERTa training on STS-B

Since we are training on different languages, we use the XLM-RoBERTa model as the base model and finetune it on different STS-B tasks on English, Spanish, Polish and Russian languages. For finetuning we use AdamW optimizer with a learning rate of $2e-5$ which is linearly annealed, with 30% of total steps for warmup, for 3 epochs. We use a batch size of 8 and regularize the training with a weight decay of 0.01. The fine-tuning takes around 5 hours in total for all random seeds.

Figure 7 *Right* shows Spearman's correlation of the predicted sentence similarity with the ground truth. The performance on English is the highest while that on Russian is the lowest. This performance across the languages correlates perfectly with the domain similarity reported in Section 4.3.

C.4 Computational resources

We used two GPUs: 2080Ti and Tesla V100, in our internal cluster.

D Additional experiments on detecting similar NNs from random NNs.

D.1 Generate random NNs with permutation

Different from the main text where we consider random NNs as untrained NNs with different initializations, here we permute the weight matrix of each layer given a pretrained ResNet-18 on ImageNet. This can preserve the parameter distributions of the pretrained ResNet while generating random functions. From Figure 8, we observe that the conclusion drawn in the main text also holds when the null distribution is constructed with permutation.

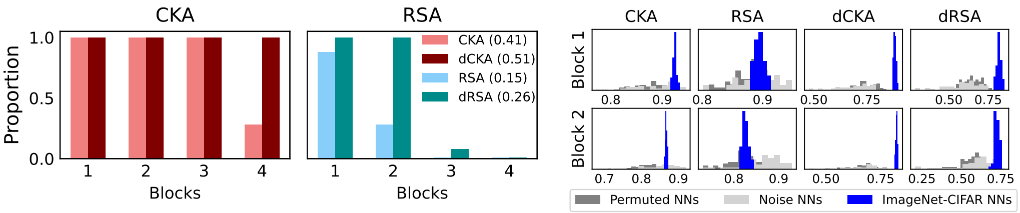

Figure 8: *Left*: Proportion of ImageNet-CIFAR ResNets pairs identified from permuted ResNets for each block. The proportions for the last four blocks are omitted as they are 0 for all metrics. *Right*: Histograms of similarities in the first two blocks with two different random NNs.

D.2 Recursive deconfounding

From Table 2 *Left*, we observe that although deconfounded similarities improve detecting semantically similar NNs from random NNs, no similarity can identify ImageNet-CIFAR pairs for deep layers. One hypothesis is that the confounding effect of input similarity cannot be approximated well with additively separable functions for deeper layers, because of the model nonlinearity added by each NN layer.

Here we consider a natural extension of deconfounding input similarity: instead of regressing out the input similarity directly (in Eq.3), we regress out the representation similarity structure from the previous layer recursively:

$$\begin{aligned} dK^{m_1}_{f_1} &= K^{m_1}_{f_1} - \hat{\beta}^{m_1}_{f_1} K^{m_1-1}_{f_1};\\ dK^{m_2}_{f_2} &= K^{m_2}_{f_2} - \hat{\beta}^{m_2}_{f_2} K^{m_2-1}_{f_2}. \end{aligned} \tag{18}$$

Although Eq.18 has the same additively separable assumption as Eq.3, but Eq.18 is easier to satisfy because it only assumes additively separable for one layer. We call the similarity generated by Eq.18 recursive deconfounded similarity.

We apply recursive deconfounded similarity, such as rdCKA and rdRSA, on the experiments of detecting similar networks from random networks described in Section 4.1. We show the comparison results between dCKA/dRSA with rdCKA/rdRSA in Table 3, where we observe a marginal

Table 3: Proportion of ImageNet-CIFAR ResNets pairs identified from random ResNets.

Block	dCKA	rdCKA	dRSA	rdRSA
1	1.0, 1.0	1.0, 1.0	1.0, 1.0	1.0, 1.0
2	1.0, 1.0	1.0, 1.0	1.0, 1.0	0.44, 0.98
3	1.0, 1.0	1.0, 1.0	0.0.0, 0.08	1.0, 1.0
4	1.0, 1.0	0.42, 1.0	0.0, 0.0	0.08, 0.02
5	0.0, 0.04	0.0, 0.78	0.0, 0.0	0.18, 0.0
6	0.0, 0.02	0.0, 0.04	0.0, 0.0	0.28, 0.4
7	0.0, 0.0	0.0, 0.0	0.0, 0.0	0.0, 0.0
8	0.0, 0.0	0.18, 0.12	0.0, 0.0	0.02, 0.18
Average	**0.5**, 0.51	0.45, **0.62**	0.25, 0.26	**0.38, 0.45**

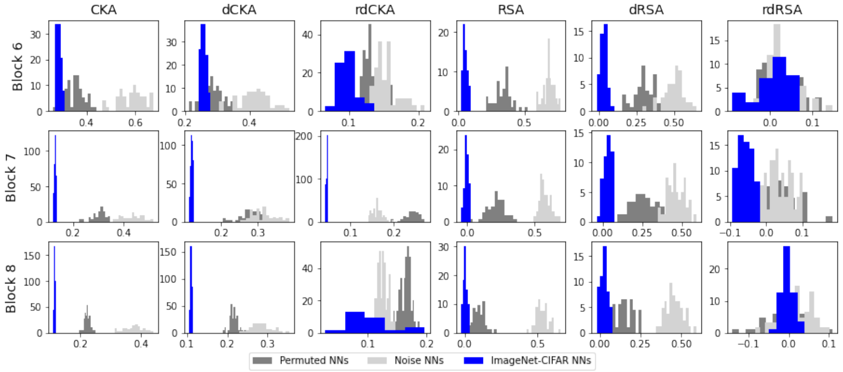

Figure 9: Histogram of each similarity measure for the last three blocks.

improvement from dCKA to rdCKA but a significant improvement from dRSA to rdRSA. However, the proportion of identified similar networks is still low for deeper layers. Thus, we consider that ImageNet and CIFAR-10 learn different high-level representations because they contain different classes of images, as mentioned in the main text.

We visualize the histogram of each similarity measure for the last 3 blocks in Figure 9. We observe that CKA/dCKA and RSA/dRSA are much smaller than two null distributions, while rdCKA/rdRSA can have a similar level as the corresponding null distribution.

E Consistency of in-domain NN functional similarities.

In Section 4.2, we test the consistency of different NN similarities across 19 different domains. In this section, we verify if NN similarities are consistent across different input samples from the same domain. We repeat the same procedure described in Section 4.2, except that we calculate the similarity $s(f_i, f^*)$ on 20 different sets of examples sampled from the same domain, i.e., the CIFAR-10 test set, instead of 19 different OOD domains.

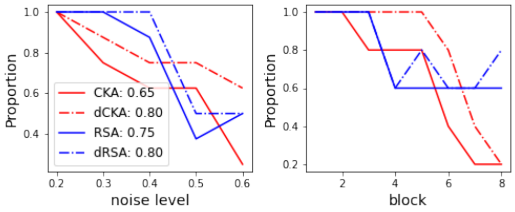

Figure 10: **Proportion of identified similar NNs across different samples from the same domain.** We observe that the deconfounded similarity can still identify more similar models compared with the corresponding original similarity.

We show the results in Figure 10, where we observe that deconfounding can improve CKA/RSA with different inputs from the same domain, but the improvement is marginal compared with cross-domain experiments: 23% for CKA (from 0.65 to 0.8) and 7% for RSA (from 0.75 to 0.8), although the proportion of identified similar NNs are much larger than the cross-domain examples for all similarity measures.

F Layer-wise CKA and dCKA on DomainNet

We show the results of the DomainNet transfer learning tasks (Section 4.3) in Figure 11. We observe that dCKA gives more consistent ranking w.r.t. the domain similarity in middle blocks.

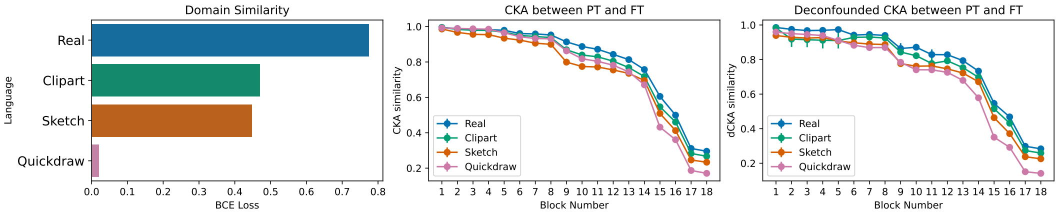

Figure 11: **dCKA adjusts for transfer learning under domain shift.** *Left* shows the ground truth domain similarity (between ImageNet and other domains in DomainNet) as measured by test binary cross entropy (BCE) loss of the cross-domain classifier. We plot the CKA (*Center*) and dCKA (*Right*) between the pretrained ResNet-50 model and models finetuned for different languages on the DomainNet tasks. We observe that dCKA is better correlated with the domain similarity than CKA.

G Consistency of dCKA for NNs trained with different initialization

One advantage of original CKA is that it can reveal consistent relationships between layers of NNs trained with different initializations, while other representation similarity measures, such as PWCCA and Procrustes, can not [19, 15]. Here we show that dCKA has a similar behavior as CKA when studying the similarity between representations of NNs with different initializations given one domain.

As in [15], we first take 5 ResNet-18 models trained on CIFAR-10 dataset with different initializations. For one model (model i), we compute the similarity between every pair of layers of the model. We then choose another model with a different initialization (model j), and compute the similarity between every layer of model i and every layer of model j. We average the results for five different models. We show the results in Figure 12, where we observe that for both CKA and dCKA and for each layer of the model i, the most similar layer in model j is the same corresponding layer. Hence, similarly to CKA, dCKA can identify consistent relationships of layers between different networks.

H Licenses

The source code and pretrained model for XLM-RoBERTa are available at `https://huggingface.co/transformers/v4.9.2/model_doc/xlmroberta.html` under the Apache License 2.0.

The source code of Distil-RoBERTa are available at `https://huggingface.co/distilroberta-base` under the Apache License 2.0.

The source code and pretrained model of ResNets on ImageNet are available at `https://pytorch.org/hub/pytorch_vision_resnet/` under license `https://github.com/pytorch/pytorch/blob/master/LICENSE`.

The source code of EfficientNet-50 are available at `https://pytorch.org/hub/nvidia_deeplearningexamples_efficientnet/` under license `https://github.com/pytorch/pytorch/blob/master/LICENSE`.

The DomainNet dataset is available from `http://ai.bu.edu/M3SDA/#dataset` with copyright information on the same website.

The STS-B dataset is available from `https://paperswithcode.com/dataset/sts-benchmark` under CC-BY-SA.

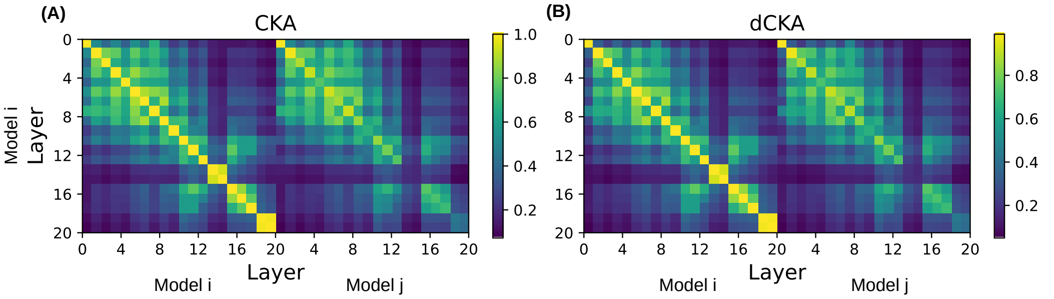

Figure 12: **CKA (A) and dCKA (B) between layers**. *Left* panel: layers are from the same NN. *Right* panel: layers are from NNs trained with different initializations. Like CKA, dCKA can reveal consistent relationships between layers of NNs trained with different initializations.

The CIFAR-10 dataset is available from `https://www.cs.toronto.edu/~kriz/cifar.html` under MIT license.

The CIFAR-10-C dataset is available from `https://zenodo.org/record/2535967#.YoThrKjP2Uk` under CC-BY 4.0.