Rebuttal: Deep Model Transferability from Attribution Maps (Paper ID 3328)

We would like to thank the AC and all the reviewers for the constructive comments, and would like to address them as follows. Due to the page limit, we provide short responses but will include more details in the final version.

To Reviewer #1

Q1: How the performance will be affected if the attribution maps are quantified?
A1: We evaluate the proposed method using different number of bits, $\{1, 2, 4, 8, 16, 32\}$, to represent an element in the attribution maps. Spearman correlations between the result (affinity matrix) of 32 bits and those of $\{1, 2, 4, 8, 16\}$ bits are $\{0.56, 0.71, 0.85, 0.96, 0.99\}$, respectively. It can be seen that, with appropriately fewer bits the proposed method also works well; however, too few (1 or 2) bits may largely affect the result.
Q2: What is the principle for choosing the layer for computing attribution maps?
A2: All taskonomy models follow the encoder-decoder architecture. For these models, we choose the output of the encoder to compute the attribution maps. Non-taskonomy models, in fact, can also be viewed as encoder-decoder ones. For example, in classification models, the convolution layers can be viewed as the encoder and the fully connected layers as the decoder. The attribution maps are thus also computed with respect to the output of the encoder.
Q3: How the attribution maps change as the layers go deeper? Please provide more results in the supplement material.
A3: Thanks. Shallow layers produce attribution maps where relevance scores are distributed uniformly; as the layers go deeper, the attribution maps tend to focus more on task-relevant regions. More details will be added to the revision.

To Reviewer #2

Q1: It will be better if the authors could show and analyze some bad cases.
A1: Thanks for the comment. For the target task Class Places, the obtained order of source tasks from our method is in fact not so similar to that produced by taskonomy. The main reason may be that, in taskonomy, most models are trained for 2D, 3D, or low dimensional geometric tasks that are very different from Class Places. These tasks may produce comparable performances when transferred to Class Places, so that the rank of source tasks is not really meaningful.
Q2: The authors should provide more discussions on the rationale behind the fact that, the proposed method works well even the probe data are quite different from the training data of the trained models.
A2: Thanks. Our basic assumption is, trained models of similar tasks should produce similar attribution maps given the same data that are randomly sampled, even if these data are from a different domain from the training data. We will provide more discussion on this issue in the final version.
Q3: What's the advantage of the proposed method over SVCCA?
A3: The main advantage is efficiency. In SVCCA, we need to compute the correlation between the features of every two tasks. However, in our method, we only need to project the pre-trained models into a common model space. The task affinity matrix is derived from the distance between points in this space, in a plug-and-play fashion.

To Reviewer #4

Q1: The usage of mathematical symbols should be consistent.
A1: Thanks for pointing out this issue. We will revise the inconsistencies in the final version.
Q2: Experimental mistake? In Figure 4, according to the precision and recall curves (the higher the better), saliency is better than ϵ-LRP and gradient. However, in Line 267, a completely reversed conclusion is given.
A2: Thanks for the comments. There is indeed no mistake here. We believe the reviewer might have taken the *oracle* curve in Fig. 4 for the *taskonomy_saliency* curve, both of which have a similar color. In fact, the three saliency curves (*taskonomy_saliency*, *indoor_saliency* and *coco_saliency*) are lower than other curves except that of *random ranking*, meaning that the results are consistent with our conclusion. We will tune the curve colors to avoid such confusion.
Q3: Since all the three attribution methods are employed from previous work, it would be better to present more discussions on the relation/interpretation/understanding among Saliency, Gradient*Input, and ϵ-LRP maps.
A3: Thanks for the suggestion. In short, Saliency constructs attributions by taking the absolute value of the partial derivative of the target output with respect to the input. Gradient*Input refers to a first-order Taylor approximation of how the output would change if the input was set to zero. ϵ-LRP, on the other hand, computes the attributions by redistributing the prediction score (output) layer by layer until the input layer is reached. As suggested, we will provide more details of the three in the final version to make the paper easier to follow.
Q4: Some conclusions are kind of trivial. For instance, "all ImageNet-trained models tend to cluster together", "the same-task trained models with similar architectures tend to be more related than with dissimilar architectures", and etc. These conclusions are rather obvious since the model embeddings are calculated using gradients w.r.t. input images.
A4: Thanks for the comment. We would like the remind the reviewer that, despite all the model embeddings are computed using gradients with respect to input images, we allow the the model architectures, initializations and training processes to be different. For the same task, therefore, such different configurations may lead to different decision patterns and hence attribution maps focusing on different regions. Without the experiments we conducted, in our opinion, it might not be perfectly safe to draw the aforementioned conclusions.
Q5: No source code is provided.
A5: We promise that the source code, data, and models will be released for reproducing the results in the paper.