A TRAINING DETAILS

The models used in this paper were trained for 1 GPU day on Nvidia A40 and for 1 GPU day on Nvidia A100, in an on-premise cluster. We train models with ResNet-18 (He et al., 2016), VGG-16 (Simonyan & Zisserman, 2014) and Inception-V3 (Szegedy et al., 2016) architectures on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) datasets. All models were trained for 150 epochs with weight decay of $5e-4$, learning rate of 0.1 and learning rate step of 50. We saved the checkpoint with the best (*i.e.* highest) validation accuracy. We used SGD optimizer for all models, with SGD momentum of 0.9. For adversarially training the models, we use the 'robustness' package (Engstrom et al., 2019; Madry et al., 2019), with 100 attack steps, epsilon of 1 and step size of 0.5. We report in Table 1 the overall accuracy of each model.

Table 1: Overall in-distribution accuracy for each of the models used in our analysis.

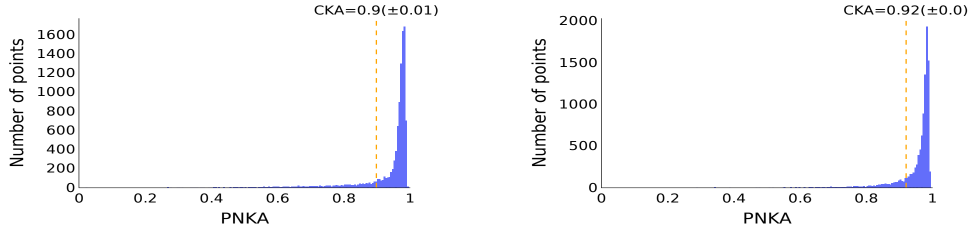
Train and Test Dataset	Training Procedure	Model	Seed	Accuracy
CIFAR-10	Standard (non-robust)	ResNet-18	0	0.9508
			1	0.9463
			2	0.9528
		VGG-16	0	0.9349
			1	0.9341
			2	0.9362
		Inception-V3	0	0.9495
			1	0.9493
			2	0.9504
	Adversarial (robust)	ResNet-18	0	0.7980
			1	0.8056
			2	0.8072
		VGG-16	0	0.7967
			1	0.7971
			2	0.7971
		Inception-V3	0	0.8108
			1	0.8000
			2	0.7983
CIFAR-100	Adversarial (robust)	ResNet-18	0	0.7713
			1	0.7729
			2	0.7722
		VGG-16	0	0.7293
			1	0.7295
			2	0.7298
		Inception-V3	0	0.7850
			1	0.7836
			2	0.7892
	Adversarial (robust)	ResNet-18	0	0.5649
			1	0.5644
			2	0.5604
		VGG-16	0	0.5238
			1	0.5136
			2	0.5187
		Inception-V3	0	0.5829
			1	0.5771
			2	0.5776

Table 2: Overall out-of-distribution accuracy for each of the models used in our analysis.

Train Dataset	Test Dataset	Training Procedure	Model	Seed	Accuracy
CIFAR-10	CIFAR-10.1	Standard (non-robust)	ResNet-18	0	0.8765
				1	0.8800
				2	0.8775
			VGG-16	0	0.8475
				1	0.8580
				2	0.8515
			Inception-V3	0	0.8855
				1	0.8870
				2	0.8850

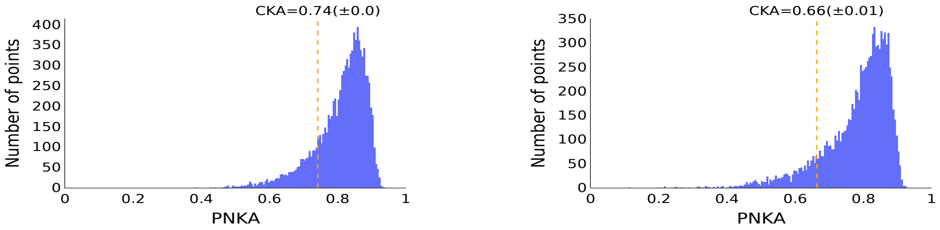
B WHY STUDY REPRESENTATION SIMILARITY AT FINER GRANULARITY

Using a pointwise representation similarity measure, we can investigate the distribution of similarity scores across points and relate it to the overall representation similarity on the entire test set. In Figure 7 we show the distribution of PNKA similarity scores for ResNet-18, VGG-16 and Inception-V3, for both CIFAR-10 and CIFAR-100 datasets. We show the average result over 3 different runs, each one containing two models with the same architecture but different random initialization. We can see that in all cases, most of the points exhibit high similarity scores, which aligns with the high CKA score obtained for the test set. However, the distribution of similarity scores is not uniform, and we observe that a few points achieve lower similarity scores, with various degrees of dissimilarity depending on the architecture and dataset being considered.



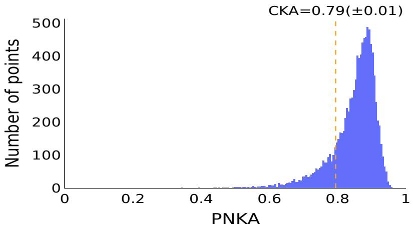
(a) VGG-16, CIFAR-10.

(b) Inception-V3, CIFAR-10.



(c) ResNet-18, CIFAR-100.

(d) VGG-16, CIFAR-100.



(e) Inception-V3, CIFAR-100.

Figure 7: Distribution of pointwise similarity scores using PNKA. Results are an average over 3 runs, each one containing two models with the same architecture but different random initialization. While most of the points are similarly represented (which explains the high aggregate CKA), some are less similarly represented.

C PROPERTIES OF PNKA

C.1 RELATIONSHIP OF PNKA WITH AGGREGATE MEASURES OF REPRESENTATION SIMILARITY

In this section, we demonstrate that the aggregate scores generated by PNKA, referred to as $\overline{\text{PNKA}}$, yield comparable CKA scores. This alignment is particularly pronounced when all L reference points correspond to the entire test set samples, *i.e.* $L = N$, as CKA computation follows this configuration. However, we can see that even using a subset of L stable and diverse points as reference yields comparable scores. Results are reported as an average (± standard deviation) over 3 runs, each one comparing two models that differ in random initialization.

Table 3: Comparison between CKA (Kornblith et al., 2019) and the aggregate version of PNKA ($\overline{\text{PNKA}}$), both with all points as reference ("all N points") or choosing a subset of L stable and diverse points ("k stable-diverse points"), where $L = 1,000$. Results are an average over 3 runs, each one with two models that only differ in their random initialization. We capture the representations of the penultimate layer (*i.e.* the layer before logits) for the analysis. We show that both measures produce similar overall scores, especially when all points are used as reference, *i.e.* $N = L$, as CKA computation follows this configuration.

Dataset	Model	CKA	$\overline{\text{PNKA}}$ (all N points)	$\overline{\text{PNKA}}$ (k stable-diverse points)
CIFAR-10	ResNet-18	0.925 (±0.005)	0.925 (±0.022)	0.958 (±0.001)
	VGG-16	0.895 (±0.013)	0.893 (±0.039)	0.941 (±0.004)
	Inception-v3	0.916 (±0.001)	0.915 (±0.023)	0.955 (±0.002)
CIFAR-100	ResNet-18	0.741 (±0.008)	0.733 (±0.033)	0.809 (±0.003)
	VGG-16	0.658 (±0.010)	0.668 (±0.049)	0.782 (±0.006)
	Inception-v3	0.798 (±0.009)	0.792 (±0.032)	0.848 (±0.005)

C.2 CHOICE OF REFERENCE POINTS

As PNKA works by comparing how a point is positioned relative to other reference points across two representation spaces, one may wonder if the reference points themselves should be required to have stable representations. For instance, in Figure 1a, computing PNKA scores using unstable (red) points as reference points might yield low similarity scores for all points. To this end, one can construct a particular case of PNKA, restricting the set of N reference points to L *stable* points. We establish that reference points in this context must adhere to two essential properties: (1) *stability*: points should remain stably positioned relative to each other, *i.e.* have high $\overline{\text{PNKA}}$ amongst themselves, and (2) *spatial diversity*: points should be well-distributed in the representation space, *i.e.* points should not be collapsed.

In the experiments, we draw $L = 1,000$ [5] reference points from the training set, *i.e.* we compute the relative position of the N test set points with respect to a subset of L stable and spatially diverse points from the training set. To define which points were going to be used as stable reference points, we compute PNKA over all the (training set) points and rank them according to the similarity score obtained. We then evaluated the properties of stability and diversity with respect to two other possible choices: (a) "Random": randomly picking points, (b) "Stable": choosing the most stable points according to the ranking, and (c) "Stable and Diverse": choosing the L/c most stable points per class of the dataset, considering c classes, *i.e.* for CIFAR-10 we choose $L = 1,000$ reference points, and since CIFAR-10 is composed of $c = 10$ classes, we selected 100 most stable instances per class.

Stability Analysis. To measure whether reference points are stable with respect to themselves, we compute the $\overline{\text{PNKA}}$ over the selected points for each of the (a), (b), and (c) possibilities. (b) "Stable" is the choice which yields higher stability, followed by (c) "Stable and diverse" and (a) "Random", respectively. This is especially the case for the CIFAR-100 dataset.

Spatial Diversity Analysis. To measure whether reference points are collapsed in the representations, we compute the L2 distance of the selected reference points with respect to themselves, for each of

---

[5] 10% of the total amount of test set points of CIFAR-10 and CIFAR-100 datasets.

Table 4: $\overline{\text{PNKA}}$ computed over the choice of reference points. The higher the $\overline{\text{PNKA}}$, the more stably positioned is the set of recene points. (a) random means randomly picking reference points; (b) stable means picking the points with the highest PNKA scores; (c) stable + diverse means picking the points with the highest PNKA scores, per class. In all cases, we are selecting 1000 reference points.

			$\overline{\text{PNKA}}$	
Dataset	Model	(a) random	(b) stable	(c) stable + diverse
CIFAR-10	ResNet-18	0.955 (± 0.002)	0.980 (±0.001)	0.974 (±0.001)
	VGG-16	0.947 (±0.009)	0.978 (±0.005)	0.971(±0.006)
	Inception-v3	0.951 (±0.002)	0.976 (±0.001)	0.972 (±0.001)
CIFAR-100	ResNet-18	0.778 (±0.003)	0.862 (±0.002)	0.844 (±0.002)
	VGG-16	0.763 (±0.006)	0.850 (±0.005)	0.831 (±0.005)
	Inception-v3	0.811 (±0.009)	0.890 (±0.008)	0.869 (±0.005)

Table 5: Average L2 distance between the points in both models, for each choice of reference points. The higher the L2 distance, the more spread the points are in the space. (a) random means randomly picking reference points; (b) stable means picking the points with the highest PNKA scores; (c) stable + diverse means picking the points with the highest PNKA scores, per class. In all cases, we are selecting 1000 reference points.

			L2 Distance	
Dataset	Model	(a) random	(b) stable	(c) stable + diverse
CIFAR-10	ResNet-18	6.632 (± 1.659)	3.620 (±2.814)	6.610 (±1.783)
	VGG-16	8.465 (±2.261)	6.398 (±3.789)	8.503 (±2.354)
	Inception-v3	6.515 (±1.599)	4.093 (±2.661)	6.362 (±1.701)
CIFAR-100	ResNet-18	15.731 (±2.073)	16.233 (±3.228)	17.222 (±2.332)
	VGG-16	19.019 (±2.486)	17.281 (±4.051)	18.839 (± 2.475)
	Inception-v3	14.949 (±2.131)	14.117 (±3.586)	14.701 (±2.029)

the (a), (b), and (c) options. We show that (c) “Stable” is the most collapsed of the options, followed by (b) “Stable and Diverse” and (a) “Random”, respectively. This is especially the case for the CIFAR-10 dataset.

Thus, we can conclude that (a) randomly picking reference points yield diverse but not stable points, *i.e.*, in most cases randomly picking reference points showed lower stability (lower PNKA score on Table 4), but high spatial diversity (higher l2 distance between reference points on Table 5). This is especially the case for CIFAR-10, a smaller dataset and less diverse dataset than CIFAR-100. Stable points (b), on the other hand, are stable but not diverse, *i.e.*, in most cases picking the most stable reference points showed higher stability (higher PNKA score on Table 4), but low diversity (lower l2 distance between reference points on Table 5). Thus, we chose the intermediate option, where we obtain stability (high PNKA score on Table 4), but also spatial diversity (higher l2 distance on Table 5).

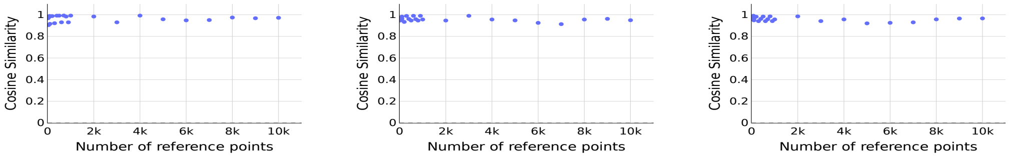
C.2.1 IMPACT OF REFERENCE POINT SELECTION IN PNKA

As defined in Section 3, PNKA scores are computed by first comparing how similarly a point is positioned relative to all the other points within each representation, and then comparing the relative position of this points across both representations. To estimate how a point is positioned relative to all the other points in a representation, one can use all the other (test set) points as reference. This general formulation draws inspiration from CKA, where each point’s position is compared to all the (test set) data points, in the same representation. However, as discussed in the paragraph titled “Computing PNKA with stable reference points” within the same Section 3, it is also possible to use different reference points when computing PNKA. The alternative choices for reference points may include the k most stable (training set) points, *i.e.* points whose positions do not change across both representations, or randomly selecting k (training set) points.

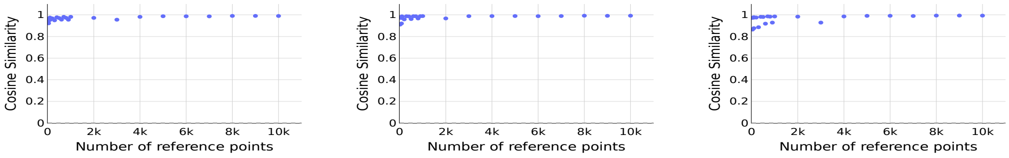
In this section, our goal is to demonstrate how the choice of reference points impacts PNKA scores. We establish as baseline the general case (with all test set points as reference), and compare this approach with the other two: (a) the k most stably represented (training set) points, and (b) k randomly chosen (training set) points. Thus, we compute the cosine similarity and pearson correlation between the PNKA scores obtained using the general (baseline) formulation, and PNKA scores with approaches (a) and (b), respectively. We systematically vary the value of k to be between 20 and 100 with an increment of 20, 100 and 1,000 with an increment of 100, and 1,000 and 10,000 (matching the number of test set points) with an increment of 1,000. The results are visualized in Figures 8 and Figures 9 for approach (a), and Figures 10 and Figures 11 for approach (b).

Our observations reveal that both the cosine similarity and Pearson correlation exhibit high values across all architectures (ResNet-18, VGG-16, Inception-V3) and datasets (CIFAR-10 and CIFAR-100), especially when $k > 1,000$. Thus, PNKA distribution is not highly impacted by the choice of reference points.

Most stable k (training set) points as reference

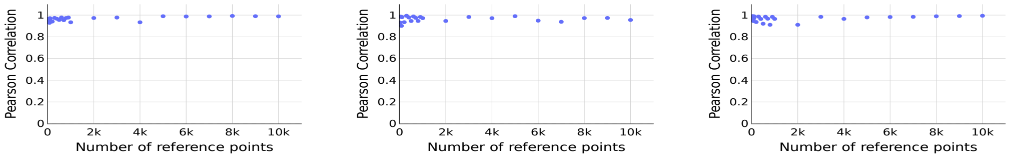


(a) ResNet-18, CIFAR-10 (b) VGG-16, CIFAR-10 (c) Incep-V3, CIFAR-10

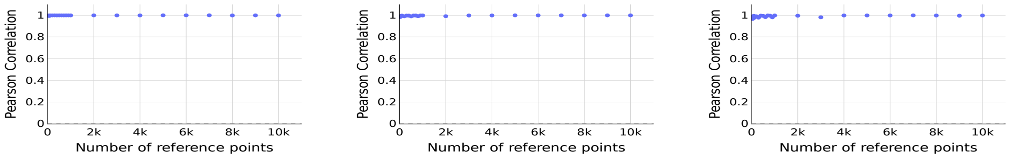


(d) ResNet-18, CIFAR-100 (e) VGG-16, CIFAR-100 (f) Incep-V3, CIFAR-100

Figure 8: Cosine similarity between PNKA scores for general case, where all (test set) points are used as reference, and PNKA scores when the k most stable (training set) points are used as reference. PNKA is averaged over 3 runs, each one containing two models trained on CIFAR-10 with different random initialization.



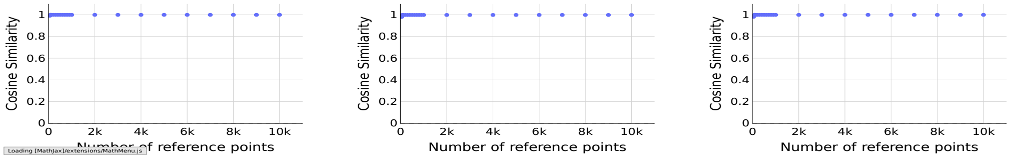
(a) ResNet-18, CIFAR-10 (b) VGG-16, CIFAR-10 (c) Incep-V3, CIFAR-10



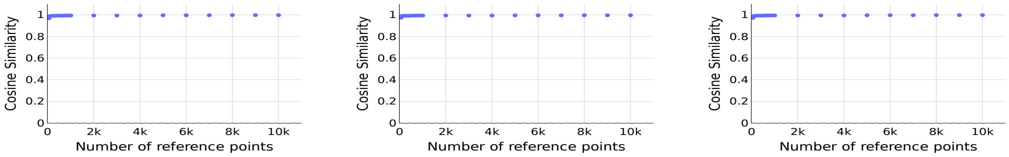
(d) ResNet-18, CIFAR-100 (e) VGG-16, CIFAR-100 (f) Incep-V3, CIFAR-100

Figure 9: Pearson correlation between PNKA scores for general case, where all (test set) points are used as reference, and PNKA scores when the k most stable (training set) points are used as reference. PNKA is averaged over 3 runs, each one containing two models trained on CIFAR-10 with different random initialization.

Random k (training set) points as reference

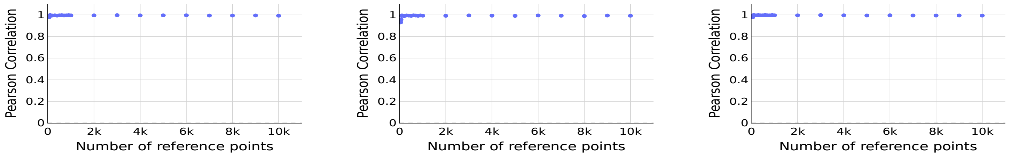


(a) ResNet-18, CIFAR-10 (b) VGG-16, CIFAR-10 (c) Incep-V3, CIFAR-10

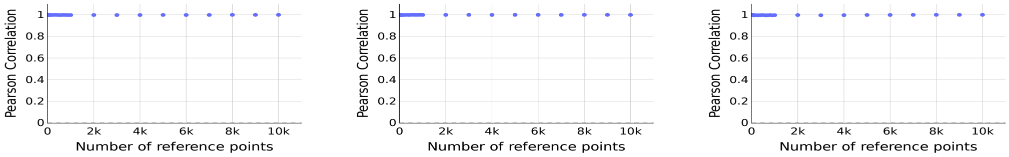


(d) ResNet-18, CIFAR-100 (e) VGG-16, CIFAR-100 (f) Incep-V3, CIFAR-100

Figure 10: Cosine similarity between PNKA scores for general case, where all (test set) points are used as reference, and PNKA scores when random k (training set) points are used as reference. PNKA is averaged over 3 runs, each one containing two models trained on CIFAR-10 with different random initialization.



(a) ResNet-18, CIFAR-10 (b) VGG-16, CIFAR-10 (c) Incep-V3, CIFAR-10



(d) ResNet-18, CIFAR-100 (e) VGG-16, CIFAR-100 (f) Incep-V3, CIFAR-100

Figure 11: Pearson correlation between PNKA scores for general case, where all (test set) points are used as reference, and PNKA scores when random k (training set) points are used as reference. PNKA is averaged over 3 runs, each one containing two models trained on CIFAR-10 with different random initialization.

C.3 PROOF OF INVARIANCES

C.3.1 INVARIANCE TO ORTHOGONAL TRANSFORMATIONS

Proof. It suffices to show that

$$\begin{aligned} K(YQ) &= YQ(YQ)^\top \\ &= YQQ^\top Y^\top \\ &= YQQ^{-1}Y^\top \\ &= YY^\top \\ &= K(Y) \end{aligned}$$

Here we have used that for an orthogonal matrix Q, $Q^\top = Q^{-1}$. By substituting $K(YQ)$ and $K(ZR)$ in $\text{PNKA}(YQ, ZR, i) = \cos(K(YQ)_i, K(ZR)_i)$ with $K(Y)$ and $K(Z)$, respectively, we obtain $\text{PNKA}(YQ, ZR, i) = \text{PNKA}(Y, Z, i)$. Thus, PNKA is invariant to orthogonal transformations. □

C.3.2 INVARIANCE TO ISOTROPIC SCALING

Proof. Note that because of the bilinearity of the dot-product, we have $K(\alpha Y)_i = \left[(\alpha Y)(\alpha Y)^\top\right]_i = \alpha^2 K(Y)_i$. By substituting into PNKA, we get

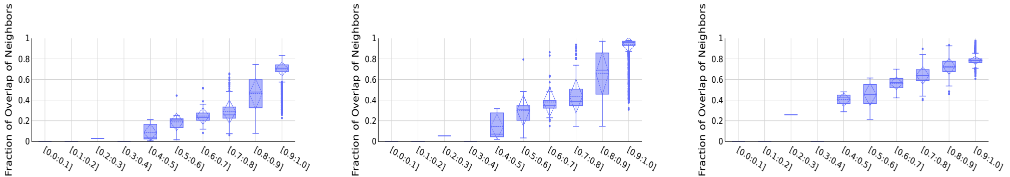
$$\begin{aligned} \text{PNKA}(\alpha Y, \beta Z, i) &= \frac{K(\alpha Y)_i^\top K(\beta Z)_i}{||K(\alpha Y)_i||_2 ||K(\beta Z)_i||_2} \\ &= \frac{\alpha^2 K(Y)_i^\top \beta^2 K(Z)}{||\alpha^2 K(Y)_i||_2 ||\beta^2 K(Z)_i||_2} \\ &= \frac{\alpha^2 K(Y)_i^\top \beta^2 K(Z)}{\alpha^2 ||K(Y)_i||_2 \beta^2 ||K(Z)_i||_2} \\ &= \text{PNKA}(Y, Z, i). \end{aligned}$$

Thus, PNKA is invariant to isotropic scaling. □

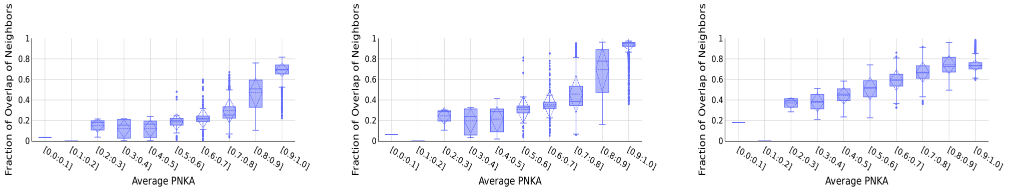
C.4 RELATION OF PNKA AND THE OVERLAP OF NEIGHBORS

We empirically show that if PNKA score of point i is higher than that of j, then there is a higher chance that i's nearest neighbors in representations Y and Z overlap more than those of j. To show this, we train two models that only differ in their random initialization and compute their representation similarity on the test set. We use the penultimate layer (*i.e.*, the layer before logits) for the analysis. For each model, we determine a point's k nearest neighbors by ranking a point's representation distance (via *cosine similarity*) to every other point in that representation. We then compute the fraction of those two sets of k neighbors that intersect.

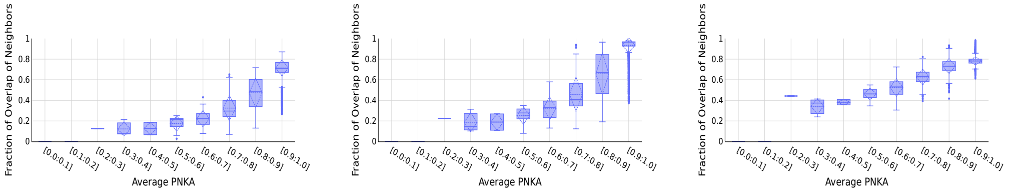
In the following plots we depict the relationship between PNKA similarity scores (x-axis) and the fraction of overlapping k nearest neighbors of each group of point (y-axis), *i.e.* 1 means all k nearest neighbors are shared between both representations. We report the analysis on CIFAR-10 (Figure 12) and CIFAR-100 (Figure 13), for ResNet-18 (He et al., 2016), VGG-16 (Simonyan & Zisserman, 2014) and Inception-V3 (Szegedy et al., 2016), for different k values, up to $k = 20\%$ of the dataset size. All the results are reported over 3 runs. In all cases, we see a relationship between higher PNKA scores for a group of points, and a higher overlap of nearest neighbors across representations, indicating that PNKA captures how similar the neighborhoods of the points are. However, we note that the results depend on the k being considered.



(a) ResNet-18, k=500 (b) ResNet-18, k=1000 (c) ResNet-18, k=2000

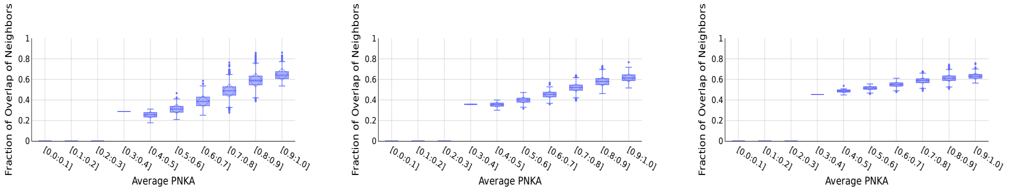


(d) VGG-16, k=500 (e) VGG-16, k=1000 (f) VGG-16, k=2000

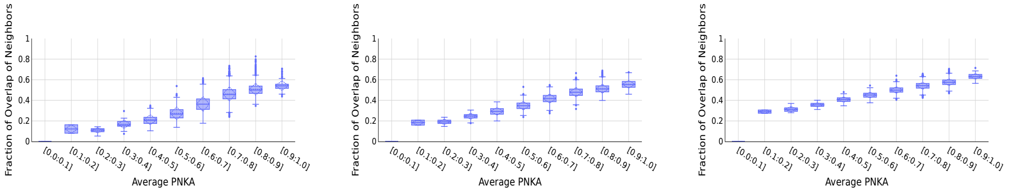


(g) Inception-V3, k=500 (h) Inception-V3, k=1000 (i) Inception-V3, k=2000

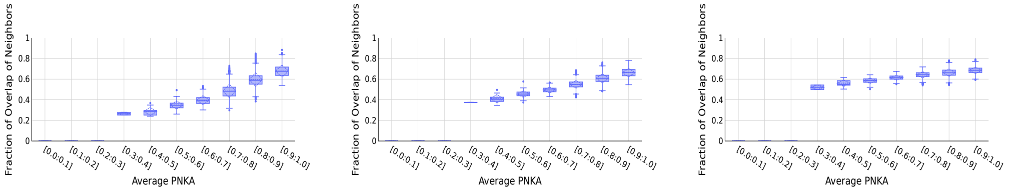
Figure 12: PNKA captures the overlap of k nearest neighbors between two representations, *i.e.*, the higher PNKA scores, the higher the fraction of overlapping neighbors. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 (Krizhevsky et al., 2009) dataset with the same architecture but different random initialization.



(a) ResNet-18, k=500 (b) ResNet-18, k=1000 (c) ResNet-18, k=2000



(d) VGG-16, k=500 (e) VGG-16, k=1000 (f) VGG-16, k=2000



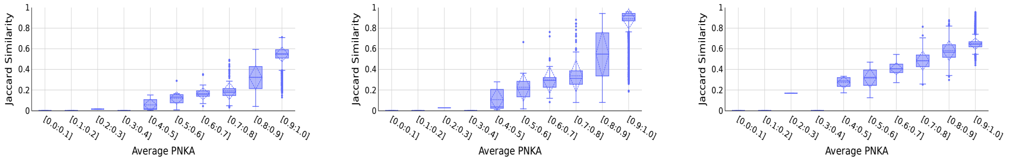
(g) Inception-V3, k=500 (h) Inception-V3, k=1000 (i) Inception-V3, k=2000

Figure 13: PNKA captures the overlap of k nearest neighbors between two representations, *i.e.*, the higher PNKA scores, the higher the fraction of overlapping neighbors. Results are an average over 3 runs, each one containing two models trained on CIFAR-100 (Krizhevsky et al., 2009) dataset with the same architecture but different random initialization.

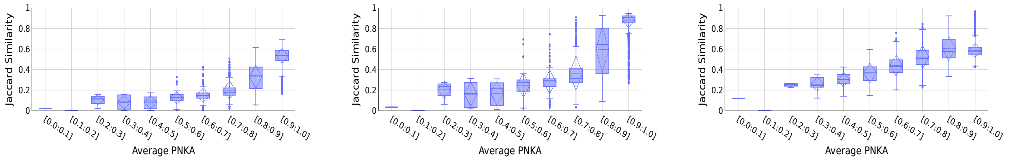
C.5 RELATION OF PNKA AND JACCARD COEFFICIENT

In this section, we investigate the relation of PNKA with Jaccard similarity coefficient. Jaccard similarity coefficient is a measure used for estimating the similarity of sample sets, and it is defined as the size of the intersection divided by the size of the union of the sample sets. In the context of representation similarity, the sets are composed of the k nearest neighbors of a point, each representation. In practice, it involves first determining the k neighbors used for the computation, and then how to measure the distance of the sample with respect to the remaining samples. For the latter, cosine similarity is a common choice Klabunde et al. (2023), and it was the one implemented in this experiment.

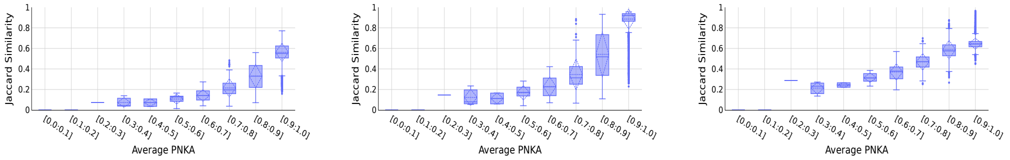
To show this relation, we train two models that only differ in their random initialization and compute their representation similarity on the test set. We use the penultimate layer (*i.e.*, the layer before logits) for the analysis. In the following plots we depict the relationship between PNKA similarity scores (x-axis) and the Jaccard similarity coefficient (y-axis). We report the analysis on CIFAR-10 (Figure 14) and CIFAR-100 (Figure 15), for ResNet-18 (He et al., 2016), VGG-16 (Simonyan & Zisserman, 2014) and Inception-V3 (Szegedy et al., 2016), for different k values, *i.e.*, the same used in the analysis of the overlap of neighbors ($k = 250, 500, 1000$). All the results are reported over 3 runs. In all cases, we see a relationship between overall high PNKA scores and high overall Jaccard similarity coefficient. However, we note that the results depend on the k being considered.



(a) ResNet-18, k=500 (b) ResNet-18, k=1000 (c) ResNet-18, k=2000

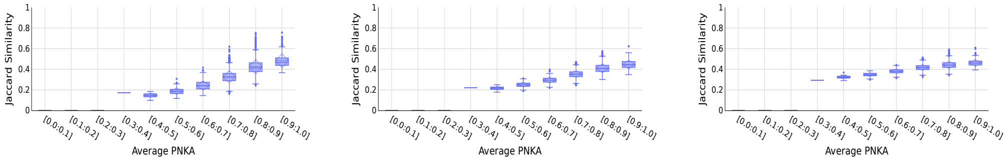


(d) VGG-16, k=500 (e) VGG-16, k=1000 (f) VGG-16, k=2000

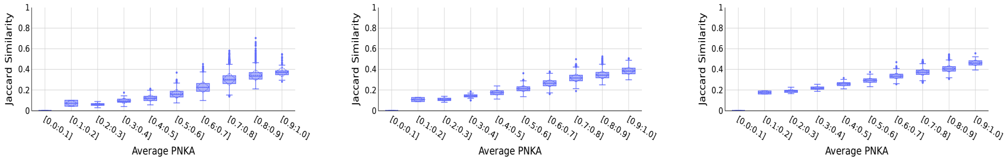


(g) Inception-V3, k=500 (h) Inception-V3, k=1000 (i) Inception-V3, k=2000

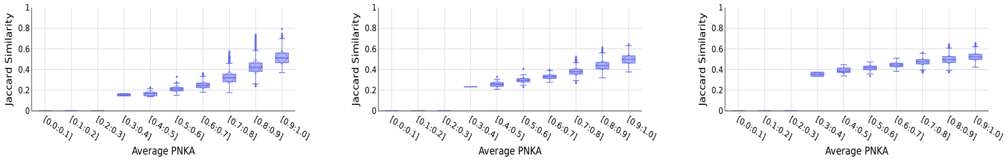
Figure 14: PNKA captures the overlap of k nearest neighbors between two representations, *i.e.*, the higher PNKA scores for a group of points, the higher the fraction of intersecting neighbors. We note, however, that the relation is dependent on the choice of k. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 (Krizhevsky et al., 2009) dataset with the same architecture but different random initialization.



(a) ResNet-18, k=500 (b) ResNet-18, k=1000 (c) ResNet-18, k=2000



(d) VGG-16, k=500 (e) VGG-16, k=1000 (f) VGG-16, k=2000



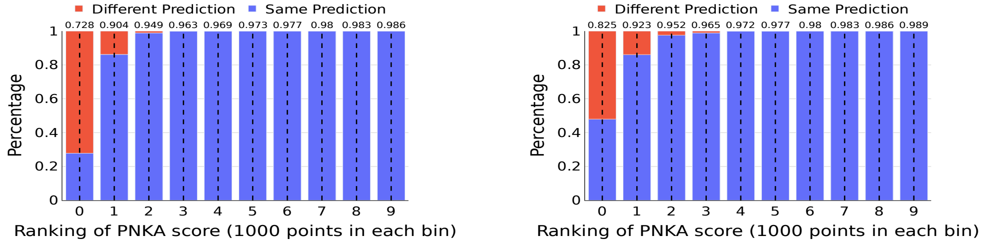
(g) Inception-V3, k=500 (h) Inception-V3, k=1000 (i) Inception-V3, k=2000

Figure 15: PNKA captures the overlap of k nearest neighbors between two representations, *i.e.*, the higher PNKA scores for a group of points, the higher the fraction of intersecting neighbors. We note, however, that the relation is dependent on the choice of k. Results are an average over 3 runs, each one containing two models trained on CIFAR-100 (Krizhevsky et al., 2009) dataset with the same architecture but different random initialization.

D USING POINTWISE ANALYSIS TO UNDERSTAND DATA INTERVENTIONS

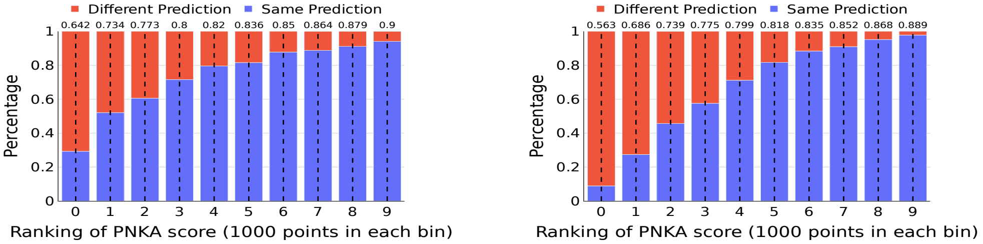
D.1 MODELS ARE MORE LIKELY TO DISAGREE ON UNSTABLE POINTS

Pointwise similarity scores allow us to connect representation similarity to other metrics of model performance. A plausible hypothesis is that inputs with low similarity scores, *i.e.* inputs represented differently by the models, will also exhibit disagreement in their predictions. In this Section, we show the same pattern for more choices of architecture and dataset. We consider the “same prediction” when all the 3 models agree on the label of the prediction. In Figure 16 we show that most of the points being dissimilarly represented are in fact the ones whose predictions the models disagree on the most.



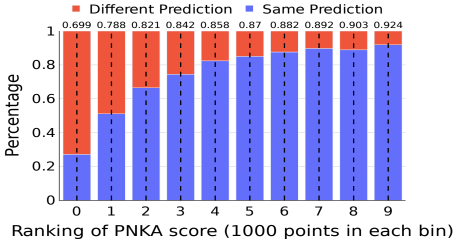
(a) VGG-16, CIFAR-10.

(b) Incep-V3, CIFAR-10.



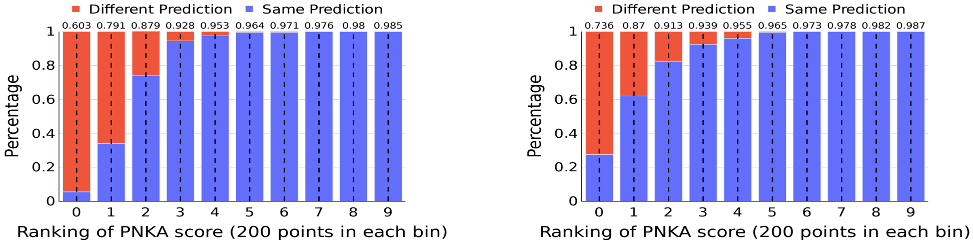
(c) ResNet-18, CIFAR-100.

(d) VGG-16, CIFAR-100.



(e) Incep-V3, CIFAR-100.

Figure 16: Percentage of instances that models agree on their predictions per group. The points have been ranked according to their similarity scores, with the left-most end (0) representing the group with the lowest scores and the right-most end (9) representing the group with the highest scores, and grouped into deciles, with each bar representing 10% of the total data points in the test set. We show results for on CIFAR-10 test set for whether models (dis)agree in their predictions. The vertical dotted line shows the aggregate scores ($\overline{\text{PNKA}}$) for that group. Most of the points that model disagree on their predictions are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 or CIFAR-100 data ssets with different random initialization.



(a) VGG-16, CIFAR-10.

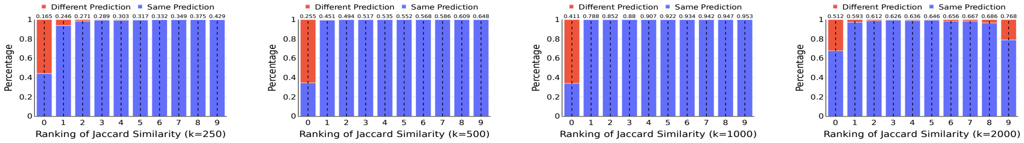
(b) Incep-V3, CIFAR-10.

Figure 17: Percentage of instances that models agree on their predictions per group. The points have been ranked according to their similarity scores, with the left-most end (0) representing the group with the lowest scores and the right-most end (9) representing the group with the highest scores, and grouped into deciles, with each bar representing 10% of the total data points in the test set. We show results for on CIFAR-10.1 test set for whether models (dis)agree in their predictions. The vertical dotted line shows the aggregate scores ($\overline{\text{PNKA}}$) for that group. Most of the points that model disagree on their predictions are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 dataset with different random initialization.

D.1.1 ANALYSIS WITH JACCARD SIMILARITY COEFFICIENT

Jaccard similarity coefficient is a measure used for estimating the similarity of sample sets, and it is defined as the size of the intersection divided by the size of the union of the sample sets. In the context of representation similarity, the sets are composed of the k nearest neighbors of a point, each representation. In practice, it involves first determining the k neighbors used for the computation, and then how to measure the distance of the sample with respect to the remaining samples. For the latter, cosine similarity is a common choice Klabunde et al. (2023), and it was the one implemented in this experiment. We run the same analysis as before (Section D.1), but using Jaccard similarity coefficient instead of PNKA. We run the experiments for four different k values (250, 500, 1000, 2000), and the results can be visualized in Figure 18, Figure 19, Figure 20, and Figure 21, for ResNet-18 and VGG-16, with both CIFAR-10 and CIFAR-100 models. From the results, we can infer that Jaccard similarity coefficient is highly influenced by the choice of k, and which k to choose from is not trivial. Moreover, the optimal k for one architecture and dataset does not generalize to other architectures and datasets. Thus, Jaccard similarity coefficient is not able to provide the same insights as PNKA.

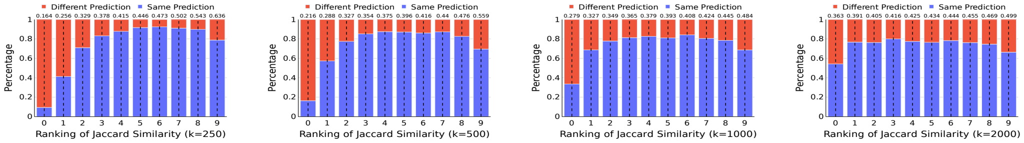
ResNet-18 models, CIFAR-10 dataset.



(a) $k = 250$. (b) $k = 500$. (c) $k = 1000$. (d) $k = 2000$.

Figure 18: Percentage of instances that models agree on their predictions per group, ranked and grouped based on their Jaccard coefficient. Results are an average over 3 runs, each one containing two ResNet-18 models trained on CIFAR-10 dataset with different random initialization.

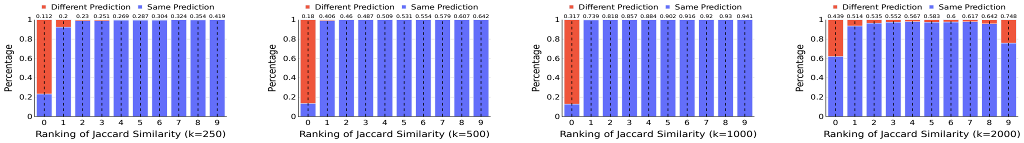
ResNet-18 models, CIFAR-100 dataset.



(a) $k = 250$. (b) $k = 500$. (c) $k = 1000$. (d) $k = 2000$.

Figure 19: Percentage of instances that models agree on their predictions per group, ranked and grouped based on their Jaccard coefficient. Results are an average over 3 runs, each one containing two ResNet-18 models trained on CIFAR-100 dataset with different random initialization.

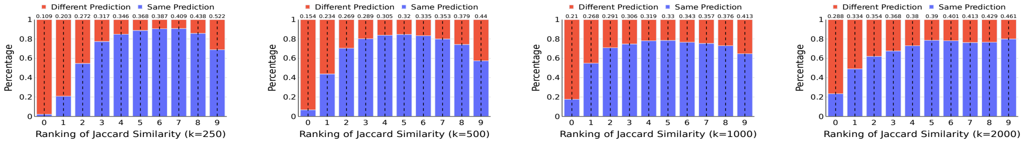
VGG-16 models, CIFAR-10 dataset.



(a) $k = 250$. (b) $k = 500$. (c) $k = 1000$. (d) $k = 2000$.

Figure 20: Percentage of instances that models agree on their predictions per group, ranked and grouped based on their Jaccard coefficient. Results are an average over 3 runs, each one containing two VGG-16 models trained on CIFAR-10 dataset with different random initialization.

VGG-16 models, CIFAR-100 dataset.

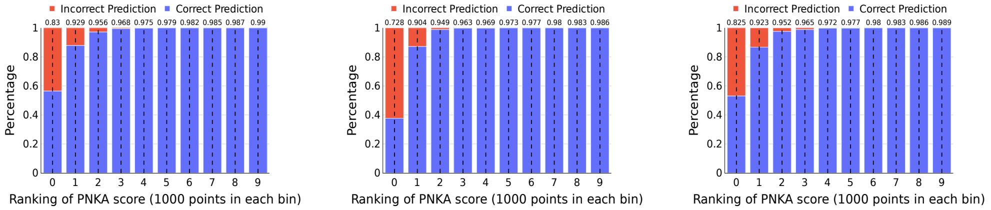


(a) $k = 250$. (b) $k = 500$. (c) $k = 1000$. (d) $k = 2000$.

Figure 21: Percentage of instances that models agree on their predictions per group, ranked and grouped based on their Jaccard coefficient. Results are an average over 3 runs, each one containing two VGG-16 models trained on CIFAR-100 dataset with different random initialization.

D.1.2 UNSTABLE POINTS ARE MORE LIKELY TO BE MISCLASSIFIED

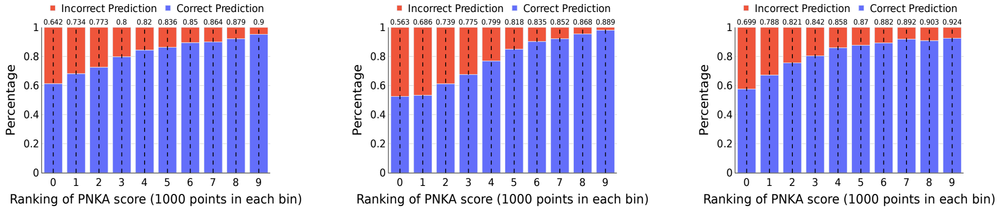
In this Section, we show that the points being dissimilarly represented are also more likely to be misclassified by the models. We consider a "correct prediction" as one where all the 3 models correctly predict the ground-truth label. Results are an average over 3 runs, each one containing two models with the same architecture but different random initialization. In Figure 22 we show that most of the points being dissimilarly represented are in fact the ones whose predictions mostly incorrect.



(a) ResNet-18, CIFAR-10.

(b) VGG-16, CIFAR-10.

(c) Incep-v3tion-V3, CIFAR-10.



(d) ResNet-18, CIFAR-100.

(e) VGG-16, CIFAR-100.

(f) Incep-v3tion-V3, CIFAR-100.

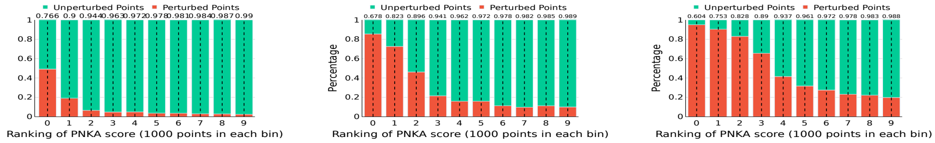
Figure 22: Percentage of instances that models correctly predict per group. The points have been ranked according to their similarity scores, with the left-most end (0) representing the group with the lowest scores and the right-most end (9) representing the group with the highest scores, and grouped into deciles, with each bar representing 10% of the total data points in the test set. The vertical dotted line shows the aggregate scores ($\overline{\text{PNKA}}$) for that group. Most of the points that models incorrectly predict are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 or CIFAR-100 datasets with different random initializations.

D.2 OUT-OF-DISTRIBUTION POINTS ARE MORE LIKELY TO HAVE UNSTABLE REPRESENTATIONS

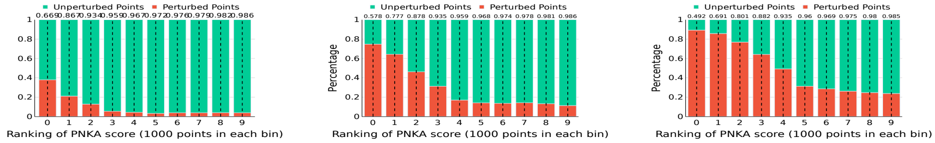
In this section, we provide additional analysis on the out-of-distribution (OOD) points. For blurring, elastic and color jitter transformations, we used the torchvision package. We set the kernel size to 9×9 and a sigma from 0.5 to 2.5 for the Gaussian blur. We set the alpha to 80 for the elastic transformation. For the color jitter, we use brightness, contrast, saturation, and hue of 0.5. We set $p\%$ of points as perturbed and $(100 - p)\%$ as non-perturbed. We show that OOD points are more likely to be dissimilarly represented, for different datasets, architectures, and p values. In each plot, we show the percentage of instances that have been perturbed or not, per group. The points have been ranked according to their similarity scores, with the left-most end (0) representing the group with the lowest scores and the right-most end (9) representing the group with the highest scores, and grouped into deciles, with each bar representing 10% of the total data points in the test set. The vertical dotted line shows the aggregate scores ($\overline{\text{PNKA}}$) for that group. We use $p\%$ perturbed and $1 - p\%$ non-perturbed points. Results are an average over 3 runs, each one containing two models with the same architecture but different random initialization.

D.2.1 BLURRING PERTURBATIONS

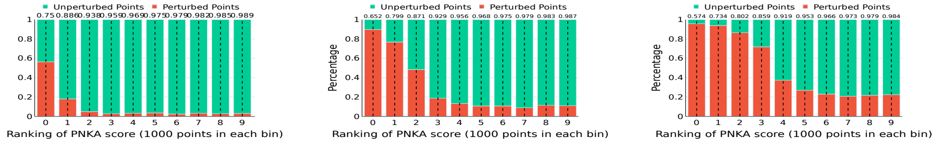
CIFAR-10



(a) R-18, p=10%. (b) R-18, p=30%. (c) R-18, p=50%.



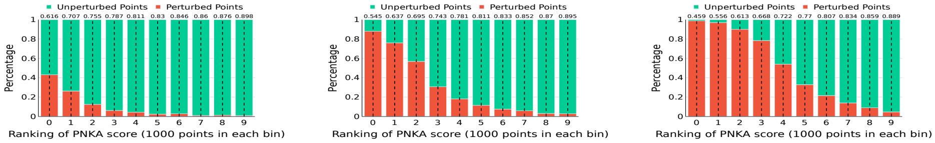
(d) VGG-16, p=10%. (e) VGG-16, p=30%. (f) VGG-16, p=50%.



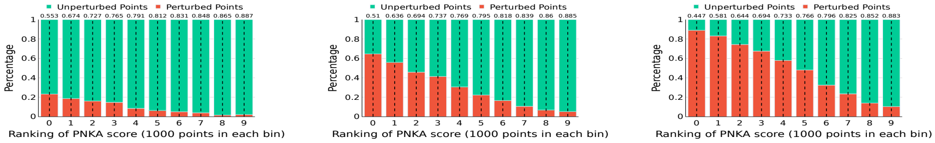
(g) Incep-v3, p=10%. (h) Incep-v3, p=30%. (i) Incep-v3, p=50%.

Figure 23: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 dataset with different random initialization.

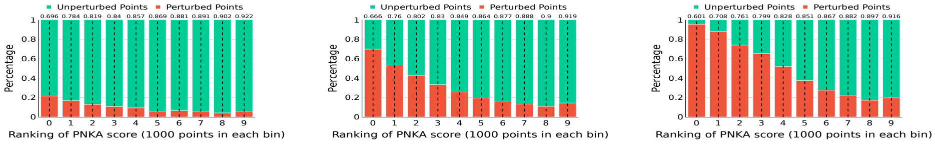
CIFAR-100



(a) R-18, p=10%. (b) R-18, p=30%. (c) R-18, p=50%.



(d) VGG-16, p=10%. (e) VGG-16, p=30%. (f) VGG-16, p=50%.

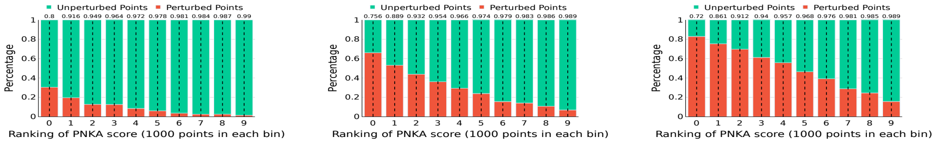


(g) Incep-v3, p=10%. (h) Incep-v3, p=30%. (i) Incep-v3, p=50%.

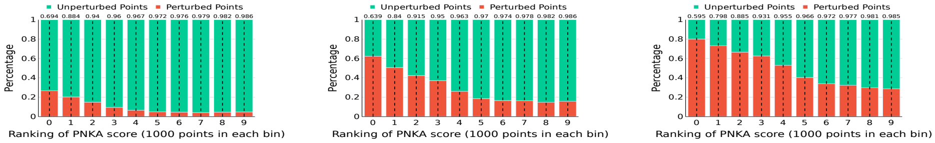
Figure 24: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-100 dataset with different random initialization.

D.2.2 ELASTIC PERTURBATIONS

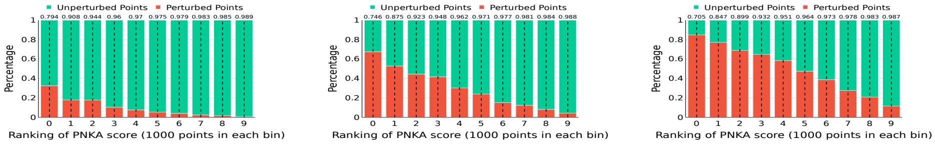
CIFAR-10



(a) R-18, p=10%. (b) R-18, p=30%. (c) R-18, p=50%.



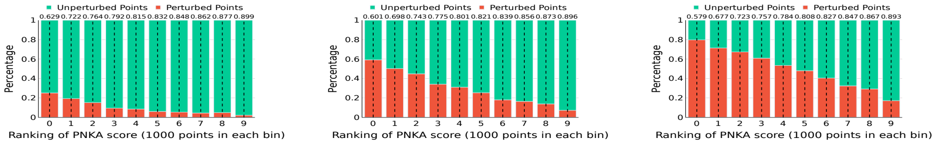
(d) VGG-16, p=10%. (e) VGG-16, p=30%. (f) VGG-16, p=50%.



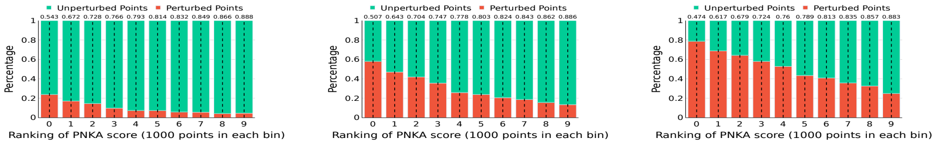
(g) Incep-v3, p=10%. (h) Incep-v3, p=30%. (i) Incep-v3, p=50%.

Figure 25: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 dataset with different random initialization.

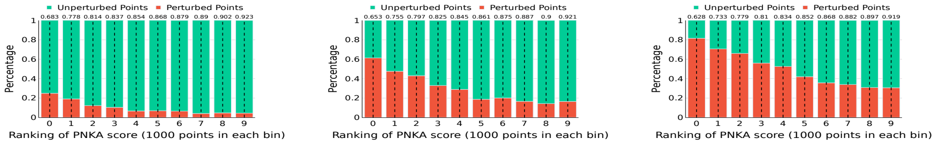
CIFAR-100



(a) R-18, p=10%. (b) R-18, p=30%. (c) R-18, p=50%.



(d) VGG-16, p=10%. (e) VGG-16, p=30%. (f) VGG-16, p=50%.

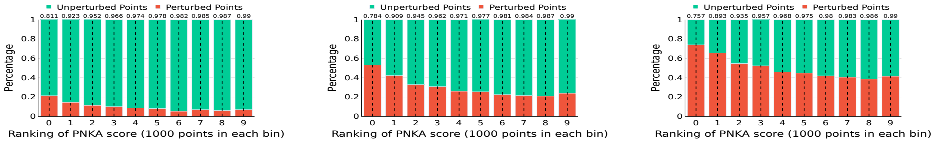


(g) Incep-v3, p=10%. (h) Incep-v3, p=30%. (i) Incep-v3, p=50%.

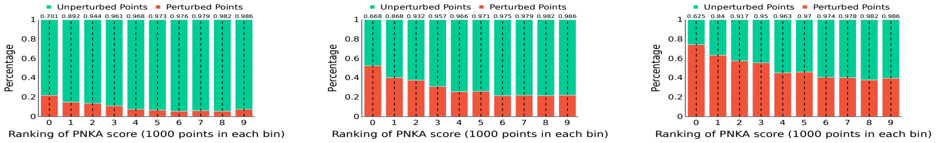
Figure 26: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-100 dataset with different random initialization.

D.2.3 COLOR JITTER PERTURBATIONS

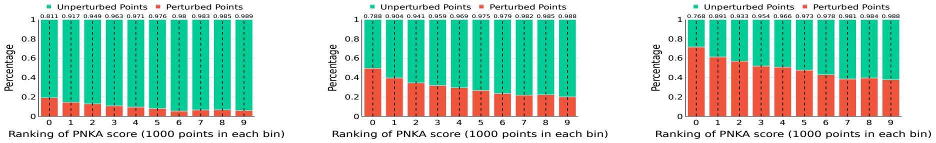
CIFAR-10



(a) R-18, p=10%. (b) R-18, p=30%. (c) R-18, p=50%.



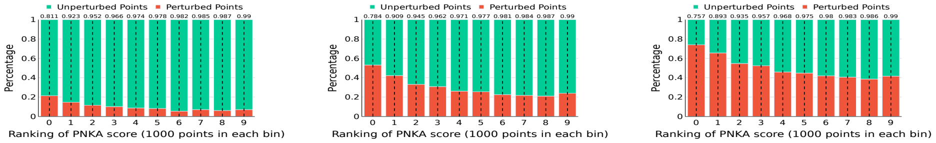
(d) VGG-16, p=10%. (e) VGG-16, p=30%. (f) VGG-16, p=50%.



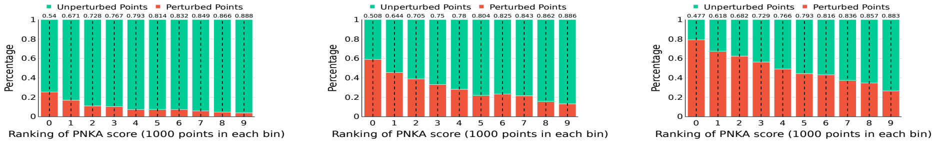
(g) Incep-v3, p=10%. (h) Incep-v3, p=30%. (i) Incep-v3, p=50%.

Figure 27: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 dataset with different random initialization.

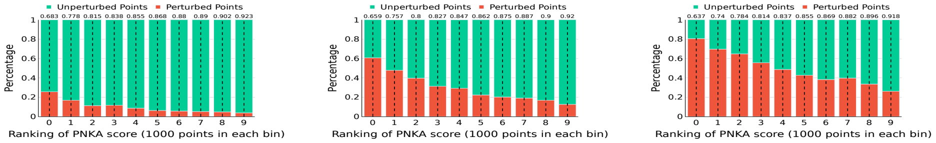
CIFAR-100



(a) R-18, p=10%. (b) R-18, p=30%. (c) R-18, p=50%.



(d) VGG-16, p=10%. (e) VGG-16, p=30%. (f) VGG-16, p=50%.

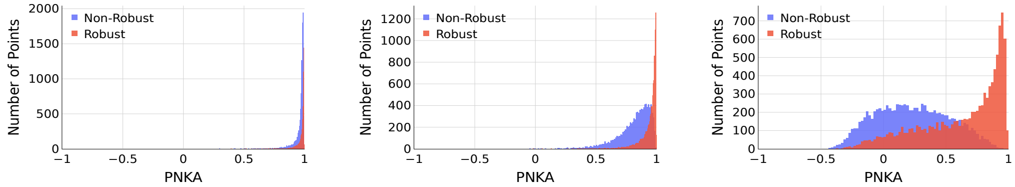


(g) Incep-v3, p=10%. (h) Incep-v3, p=30%. (i) Incep-v3, p=50%.

Figure 28: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing two models trained on CIFAR-100 dataset with different random initialization.

D.3 ROBUST MODELS ARE LESS INFLUENCED BY STOCHASTIC FACTORS

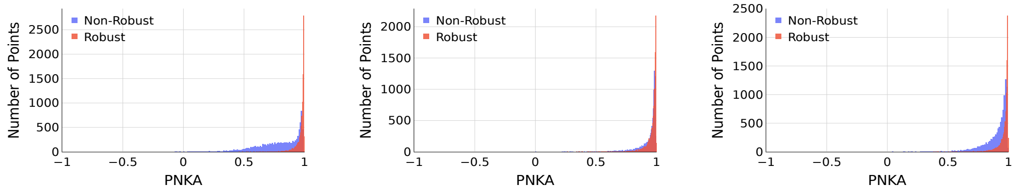
In this section, we provide more results regarding standard and robust models for various types of perturbations. In each of the plots shown below, we show the distribution of similarity scores for standard (non-robust) models (blue) and adversarially trained (robust) models (red). All plots contain results which averaged over 3 runs, each one containing two models trained on CIFAR-10 (Figure 29) or on CIFAR-100 (Figure 30) with different random initialization. The pointwise similarity scores are shown for CIFAR-10 and CIFAR-100 test sets, as well as complete random noise, and perturbed test set instances with blurring, color jitter or elastic transformation. For blurring, elastic and color jitter transformations, we used the torchvision package. We set the kernel size to 9×9 and a sigma from 0.5 to 2.5 for the Gaussian blur. We set the alpha to 80 for the elastic transformation. For the color jitter, we use brightness, contrast, saturation, and hue of 0.5. In all cases, while standard models represent (most) inputs similarly only when they are drawn from training data distribution (left-most figure), adversarially trained models represent a wide variety of out-of-distribution inputs similarly, thus indicating that these models learn more “stable” representations.



(a) R-18, C10.

(b) R-18, C100.

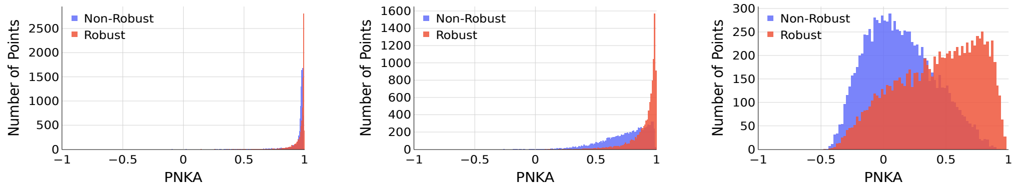
(c) R-18, noise.



(d) R-18, blur.

(e) R-18, color jitter.

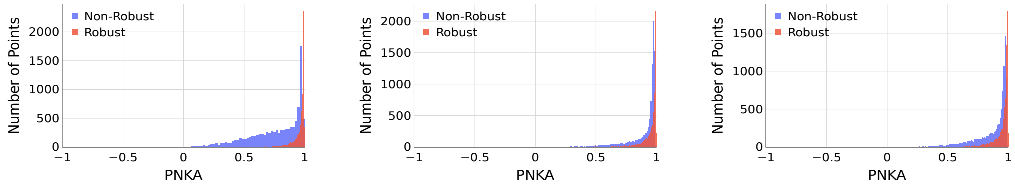
(f) R-18, elastic.



(g) VGG-16, C10.

(h) VGG-16, C100.

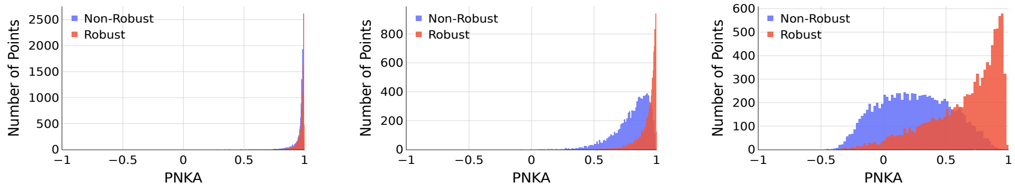
(i) VGG-16, noise.



(j) VGG-16, blur.

(k) VGG-16, color jitter.

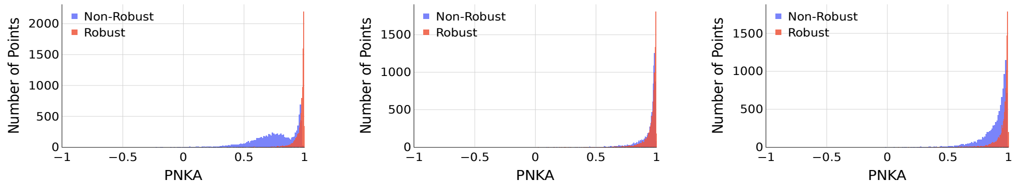
(l) VGG-16, elastic.



(m) Incep-v3-V3, C10.

(n) Incep-v3-V3, C100.

(o) Incep-v3-V3, noise.

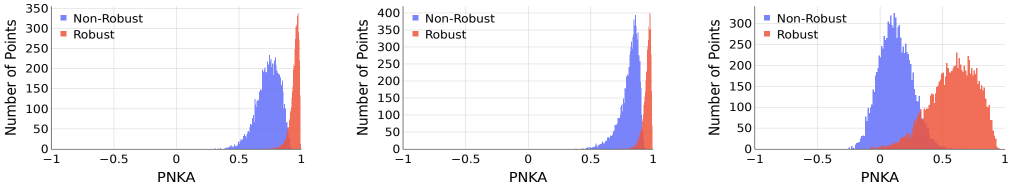


(p) Incep-v3-V3, blur.

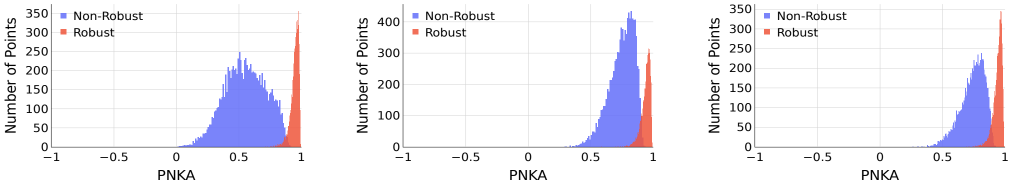
(q) Incep-v3-V3, color jitter.

(r) Incep-v3-V3, elastic.

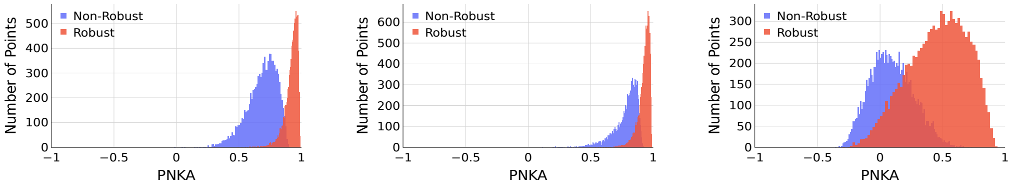
Figure 29: Distribution of similarity scores for standard (*i.e.* non-robust) and adversarially trained (*i.e.* robust) models trained on CIFAR-10 tested on in-distribution as well as other distribution shifts.



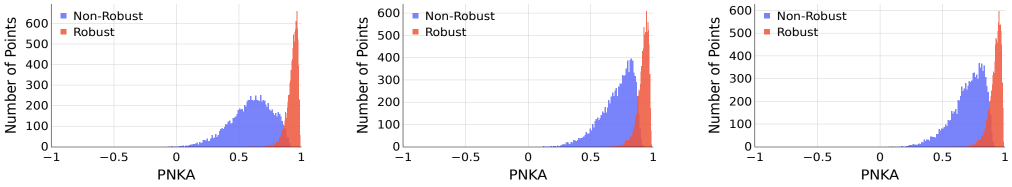
(a) R-18, C10. (b) R-18, C100. (c) R-18, noise.



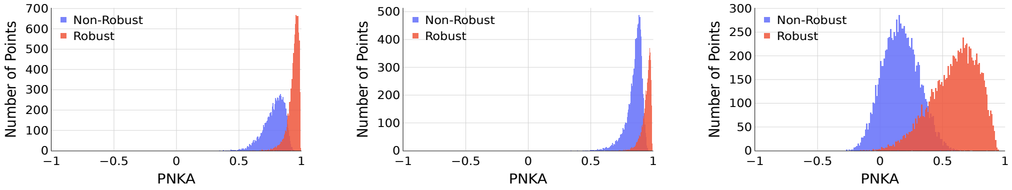
(d) R-18, blur. (e) R-18, color jitter. (f) R-18, elastic.



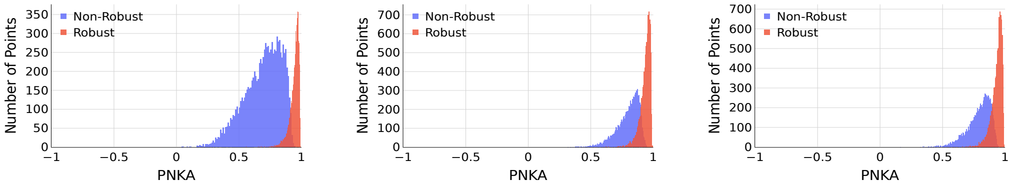
(g) VGG-16, C10. (h) VGG-16, C100. (i) VGG-16, noise.



(j) VGG-16, blur. (k) VGG-16, color jitter. (l) VGG-16, elastic.



(m) Incep-v3-V3, C10. (n) Incep-v3-V3, C100. (o) Incep-v3-V3, noise.



(p) Incep-v3-V3, blur. (q) Incep-v3-V3, color jitter. (r) Incep-v3-V3, elastic.

Figure 30: Distribution of similarity scores for standard (*i.e.* non-robust) and adversarially trained (*i.e.* robust) models trained on CIFAR-100 tested on in-distribution as well as other distribution shifts.

E USING POINTWISE ANALYSIS TO UNDERSTAND MODEL INTERVENTIONS

E.1 RESULTS ON SEMBIAS DATASET

For each of the four word pairs (a, b) in a SemBias instance, GP- and GN-Glove measure its cosine similarity with the canonical gender vector, *i.e.* $cos(\overrightarrow{a} - \overrightarrow{b}, \overrightarrow{he} - \overrightarrow{she})$. The word pair with the highest cosine similarity is selected as the "predicted" answer. If the word embeddings are correctly debiased, then the cosine similarity of the $\overrightarrow{he} - \overrightarrow{she}$ vector with the gender-definition words should be high, and the similarity with the gender-stereotype words should be low, *i.e.* the frequency of predictions for these categories should be high and low, respectively. Table 6 depicts the results for the GN- and GP-Glove (Zhao et al., 2018; Kaneko & Bollegala, 2019) methods.

Table 6: Frequency of predictions for gender relational analogies (Kaneko & Bollegala, 2019). Each column shows the frequency with which the respective word-pair category (gender-definitional, gender-stereotype, gender-neutral) is predicted as having the highest cosine similarity with the canonical gender vector $\overrightarrow{he} - \overrightarrow{she}$. The more often gender-definition words are predicted as being most gender-aligned, as opposed to gender-stereotype words, the less biased an embedding approach can be considered.

Embeddings	SemBias		
	Definition ↑	Stereotype ↓	Neutral ↓
GloVe	80.2	10.9	8.9
GN-GloVe	97.7	1.4	0.9
GP-GloVe	84.3	8.0	7.7
GP-GN-GloVe	98.4	1.1	0.5

E.2 ADDITIONAL INFORMATION ON ANALYZING GENDER CHANGES

The similarity scores from PNKA in Figure 5 (main paper) show that across all three methods, the words whose embeddings change the most are the gender-definition words. This observation, however, is not consistent with the expectation that the embeddings that should change the most are the gender-stereotypical ones, not the gender-definitional ones. The fact that the classification results for word pairs in SemBias nonetheless behave as expected suggests the hypothesis that instead of removing gender information from the gender-stereotypical word pairs, the debiasing methods might instead be amplifying the gender information in the gender-definition word pairs.

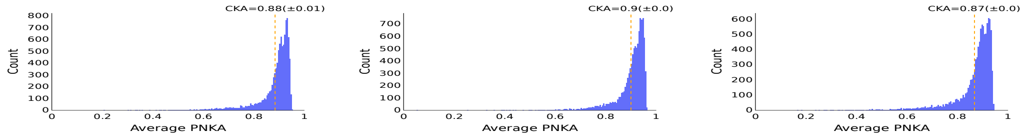
Thus, to test this hypothesis, for each embedding approach ϕ_e and word i, we project the corresponding word embeddings $w_i^{(e)} = \phi_e(i)$ into the gender vector direction $g^{(e)} = \overrightarrow{he^{(e)}} - \overrightarrow{she^{(e)}}$ and compute the projection magnitudes $p_i^{(e)} = \|g^{(e)\top} w_i^{(e)}\|_2$. The higher $p_i^{(e)}$ is, the more gender information is contained in the word embedding vector $w_i^{(e)}$. To understand how much each of the debiased embedding methods ϕ_e change the amount of gender information, relative to the original GloVe embeddings, we analyze the percentage difference in magnitude, defined as

$$\omega_i^{(e)} = \frac{p_i^{(e)} - p_i^{(glove)}}{p_i^{(glove)}}$$

$\omega_i^{(e)} = 0$ indicates that the gender information in the debiased embedding has not changed relative to GloVe, while $\omega_i^{(e)} > 0$ (or $\omega_i^{(e)} < 0$) indicates an increase in the gender information associated with i.

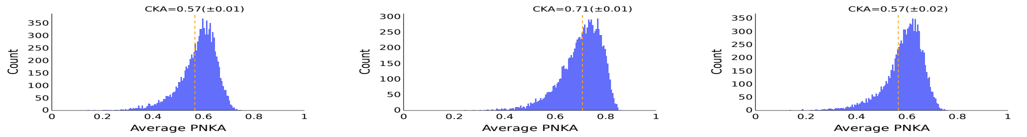
F USING POINTWISE ANALYSIS FOR MODELS WITH DIFFERENT ARCHITECTURES

Pointwise representation similarity measures can be used to also analyze models that differ in other aspects, such as architecture. Below we show the distribution of similarity scores when comparing models trained on CIFAR-10 (Figure 31) or CIFAR-100 (Figure 32). A similar analysis as conducted in Section 4.1 on the tendency of models to disagree on predictions made on unstable points is possible. We share the results for CIFAR-10 models in Figures 33, and CIFAR-100 models in Figure 34. A similar analysis regarding out-of-distribution data can also be done, as visualized in Figure 35 and Figure 37 for CIFAR-10 models, and Figure 36 and Figure 38 for CIFAR-100 models, for both the blurring and elastic perturbations, respectively.



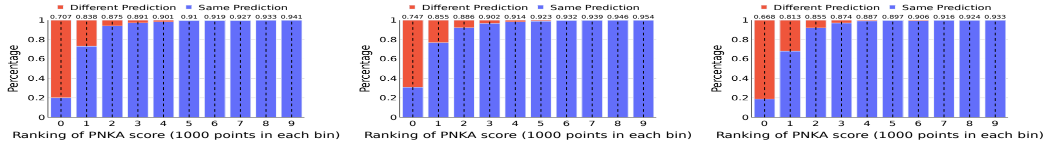
(a) R-18 × VGG-16.
(b) R-18 × Incep-V3.
(c) VGG-16 × Incep-V3.

Figure 31: Distribution of similarity scores when comparing the penultimate layer of two models trained on CIFAR-10 using different architectures.



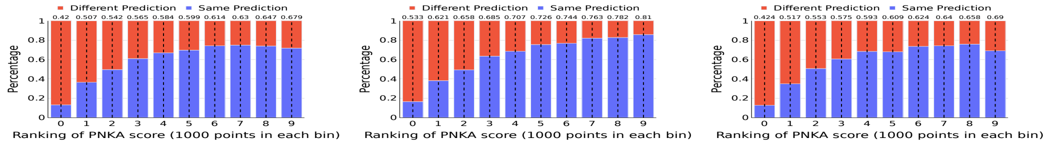
(a) R-18 × VGG-16.
(b) R-18 × Incep-V3.
(c) VGG-16 × Incep-V3.

Figure 32: Distribution of similarity scores when comparing the penultimate layer of two models trained on CIFAR-100 using different architectures.



(a) R-18 × VGG-16.
(b) R-18 × Incep-V3.
(c) VGG-16 × Incep-V3.

Figure 33: Percentage of instances that models correctly predict per group of points, ranked and grouped based on their similarity scores. Results are an average over 3 runs, each one containing two models trained on CIFAR-10 dataset with different architectures.

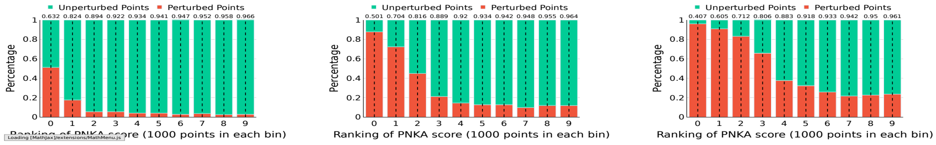


(a) R-18 × VGG-16.
(b) R-18 × Incep-V3.
(c) VGG-16 × Incep-V3.

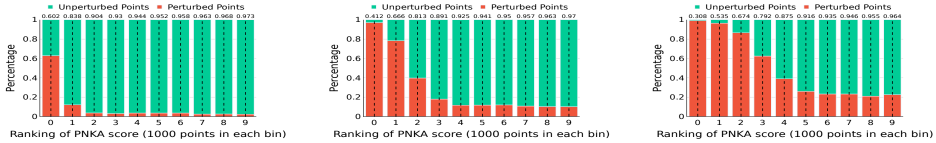
Figure 34: Percentage of instances that models correctly predict per group of points, ranked and grouped based on their similarity scores. Results are an average over 3 runs, each one containing two models trained on CIFAR-100 dataset with different architectures.

Blurring perturbations

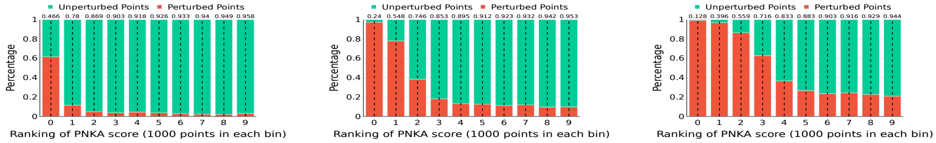
CIFAR-10



(a) R-18 × VGG-16, p=10%. (b) R-18 × VGG-16, p=30%. (c) R-18 × VGG-16, p=50%.



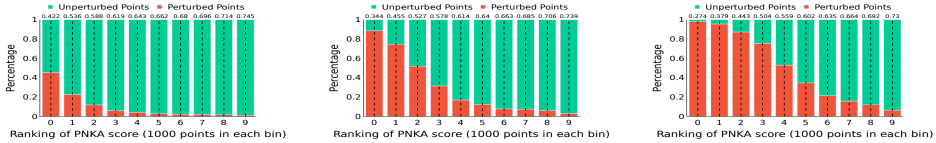
(d) R-18 × Incep-v3, p=10%. (e) R-18 × Incep-v3, p=30%. (f) R-18 × Incep-v3, p=50%.



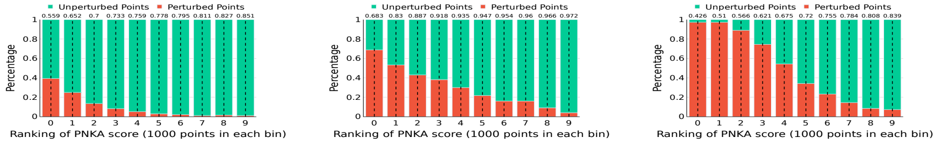
(g) VGG-16 × Incep-v3, p=10%. (h) VGG-16 × Incep-v3, p=30%. (i) VGG-16 × Incep-v3, p=50%.

Figure 35: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing models with different architectures trained on CIFAR-10 dataset.

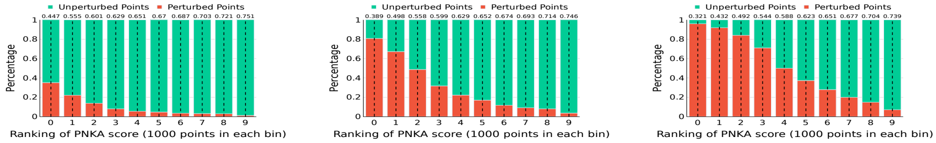
CIFAR-100



(a) R-18 × VGG-16, p=10%. (b) R-18 × VGG-16, p=30%. (c) R-18 × VGG-16, p=50%.



(d) R-18 × Incep-v3, p=10%. (e) R-18 × Incep-v3, p=30%. (f) R-18 × Incep-v3, p=50%.

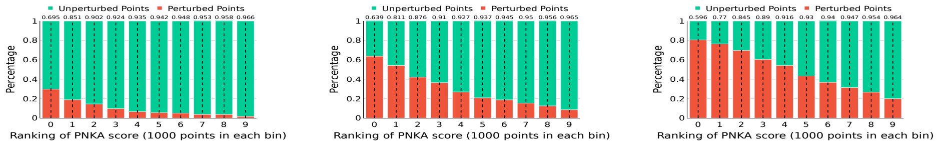


(g) VGG-16 × Incep-v3, p=10%. (h) VGG-16 × Incep-v3, p=30%. (i) VGG-16 × Incep-v3, p=50%.

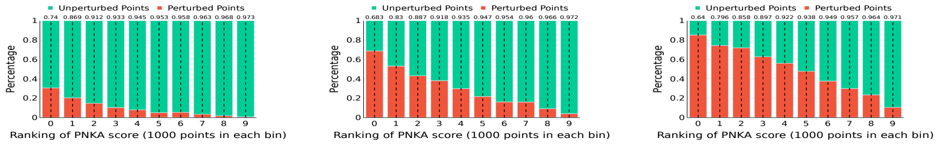
Figure 36: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing models with different architectures trained on CIFAR-100 dataset.

Elastic perturbations

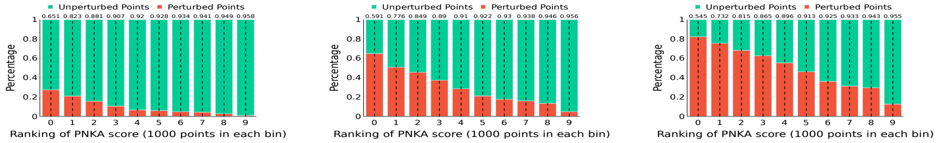
CIFAR-10



(a) R-18 × VGG-16, p=10%. (b) R-18 × VGG-16, p=30%. (c) R-18 × VGG-16, p=50%.



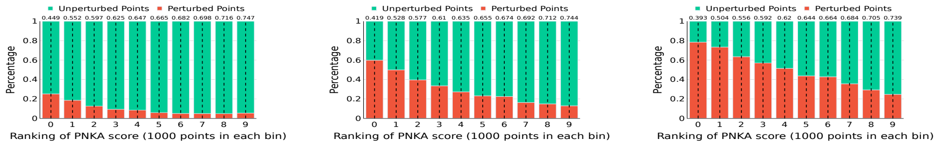
(d) R-18 × Incep-v3, p=10%. (e) R-18 × Incep-v3, p=30%. (f) R-18 × Incep-v3, p=50%.



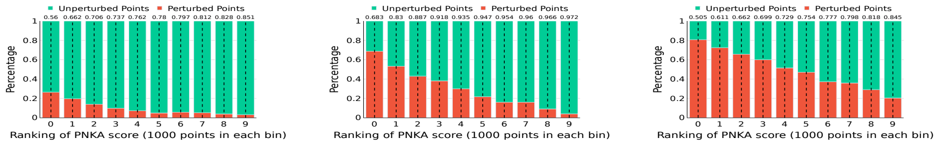
(g) VGG-16 × Incep-v3, p=10%. (h) VGG-16 × Incep-v3, p=30%. (i) VGG-16 × Incep-v3, p=50%.

Figure 37: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing models with different architectures trained on CIFAR-10 dataset.

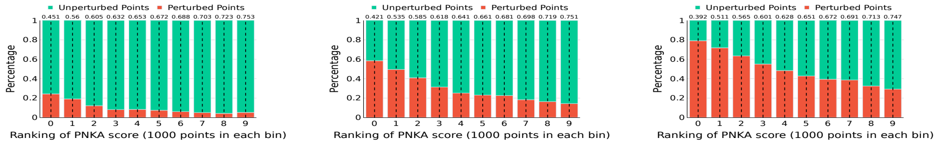
CIFAR-100



(a) R-18 × VGG-16, p=10%. (b) R-18 × VGG-16, p=30%. (c) R-18 × VGG-16, p=50%.



(d) R-18 × Incep-v3, p=10%. (e) R-18 × Incep-v3, p=30%. (f) R-18 × Incep-v3, p=50%.

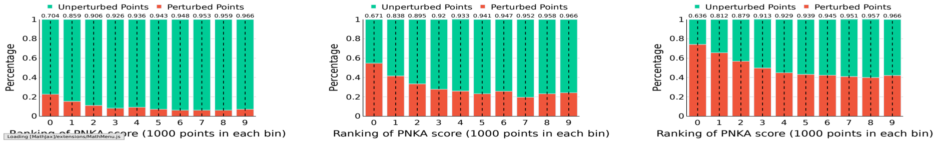


(g) VGG-16 × Incep-v3, p=10%. (h) VGG-16 × Incep-v3, p=30%. (i) VGG-16 × Incep-v3, p=50%.

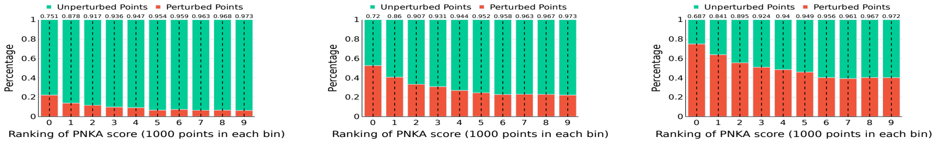
Figure 38: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing models with different architectures trained on CIFAR-100 dataset.

Color jitter perturbations

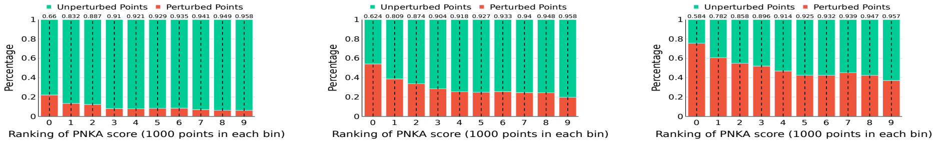
CIFAR-10



(a) R-18 × VGG-16, p=10%. (b) R-18 × VGG-16, p=30%. (c) R-18 × VGG-16, p=50%.



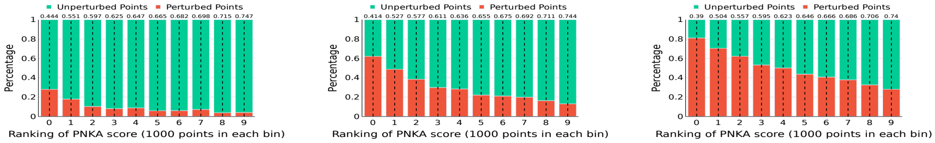
(d) R-18 × Incep-v3, p=10%. (e) R-18 × Incep-v3, p=30%. (f) R-18 × Incep-v3, p=50%.



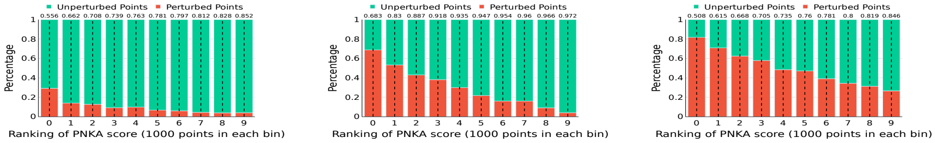
(g) VGG-16 × Incep-v3, p=10%. (h) VGG-16 × Incep-v3, p=30%. (i) VGG-16 × Incep-v3, p=50%.

Figure 39: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing models with different architectures trained on CIFAR-10 dataset.

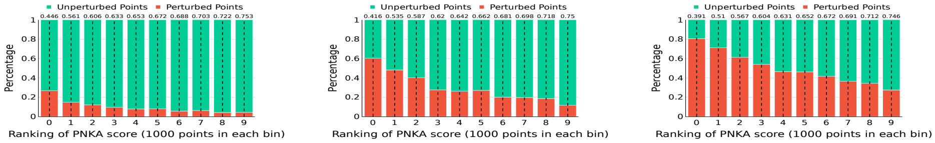
CIFAR-100



(a) R-18 × VGG-16, p=10%. (b) R-18 × VGG-16, p=30%. (c) R-18 × VGG-16, p=50%.



(d) R-18 × Incep-v3, p=10%. (e) R-18 × Incep-v3, p=30%. (f) R-18 × Incep-v3, p=50%.



(g) VGG-16 × Incep-v3, p=10%. (h) VGG-16 × Incep-v3, p=30%. (i) VGG-16 × Incep-v3, p=50%.

Figure 40: Percentage of instances that are perturbed per group of points, ranked and grouped based on their similarity scores. Most of the perturbed points are located at the lower end of the distribution. Results are an average over 3 runs, each one containing models with different architectures trained on CIFAR-100 dataset.

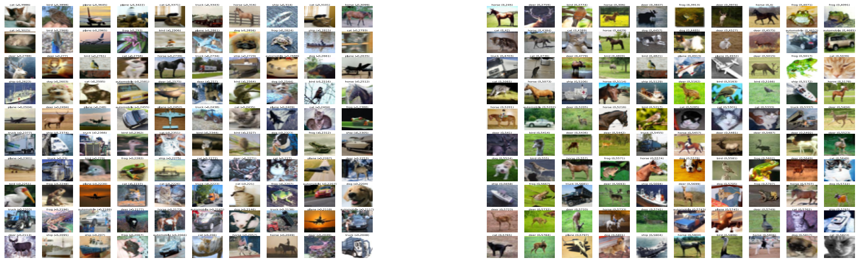
G USING POINTWISE ANALYSIS FOR DIFFERENT LAYERS OF THE SAME MODEL

Pointwise representation similarity measures can be used for analyzing representation changes within W layers of the same model. For instance, by leveraging these local measures one can inspect which points changed the most (least) its representations from one layer of the model (l) to its consecutive layer ($l + 1$), or even from one layer of the model (l) and the penultimate layer of the model ($W - 1$), *i.e.* the more (less) its representation is altered from one layer to the other, the lower (higher) its representation similarity. In this experiment, we get the representation of layers after either a block of convolutional, batch norm and relu operations, or an average pooling operation. We observe that from one layer to the other, the instances that change the most have some patterns, *e.g.* gray-scale images, or images with green or blue backgrounds. Future research could explore this direction in more depth.

G.1 COMPARING CONSECUTIVE LAYERS

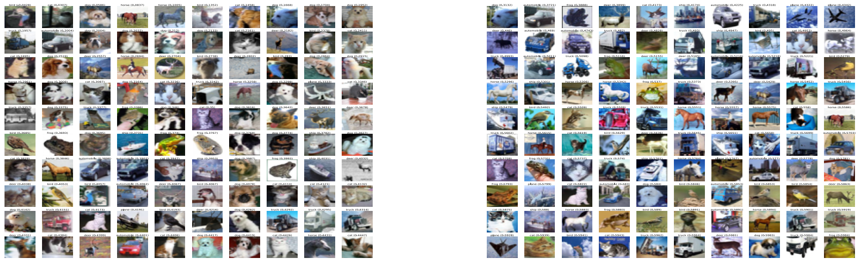
(a) Layer 1 vs layer 2.

(b) Layer 2 vs layer 3.



(c) Layer 3 vs layer 4.

(d) Layer 4 vs layer 5.



(e) Layer 5 vs layer 6.

(f) Layer 6 vs layer 7.

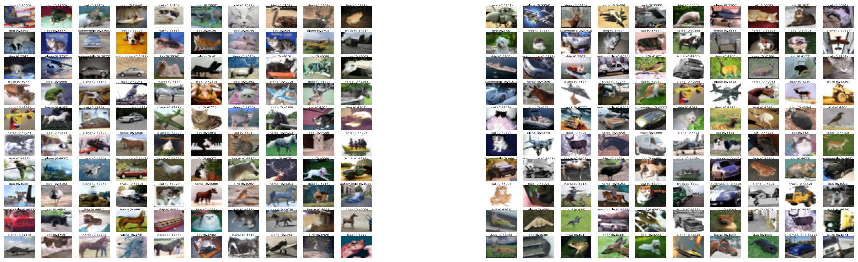
(g) Layer 7 vs layer 8.

(h) Layer 8 vs layer 9.

Figure 41: 100 most dissimilarly represented images comparing consecutive layer for the layers 0-9, *i.e.* 100 images that changed the most from only layer to the other. Model trained on CIFAR-10 training set with ResNet-18 architecture.

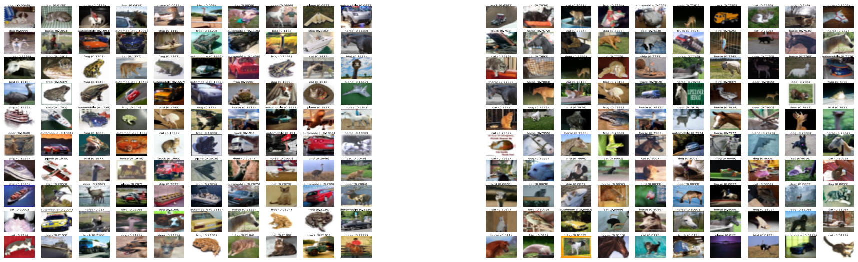
(a) Layer 9 vs layer 10.

(b) Layer 10 vs layer 11.



(c) Layer 11 vs layer 12.

(d) Layer 12 vs layer 13.



(e) Layer 13 vs layer 14.

(f) Layer 14 vs layer 15.

(g) Layer 15 vs layer 16.

(h) Layer 16 vs layer 17.

Figure 42: 100 most dissimilarly represented images comparing consecutive layer for the layers 9-17, *i.e.* 100 images that changed the most from only layer to the other. Model trained on CIFAR-10 training set with ResNet-18 architecture.

(a) Layer 16 vs layer 17.

Figure 43: 100 most dissimilarly represented images comparing consecutive layer for the layers 17-18, *i.e.* 100 images that changed the most from only layer to the other. Model trained on CIFAR-10 training set with ResNet-18 architecture.

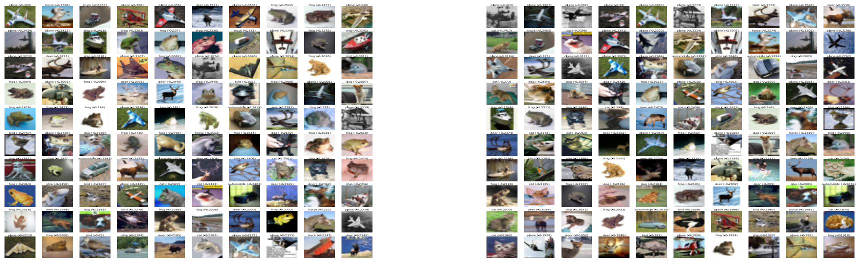
G.2 COMPARING LAYERS WITH PENULTIMATE LAYER

(a) Layer 17 vs layer 0.

(b) Layer 17 vs layer 1.

(c) Layer 17 vs layer 2.

(d) Layer 17 vs layer 3.



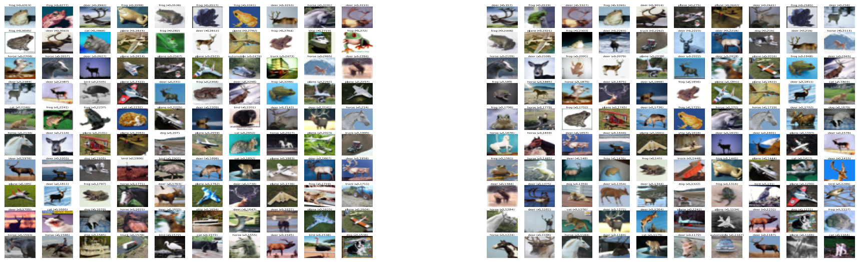
(e) Layer 17 vs layer 4.

(f) Layer 17 vs layer 5.

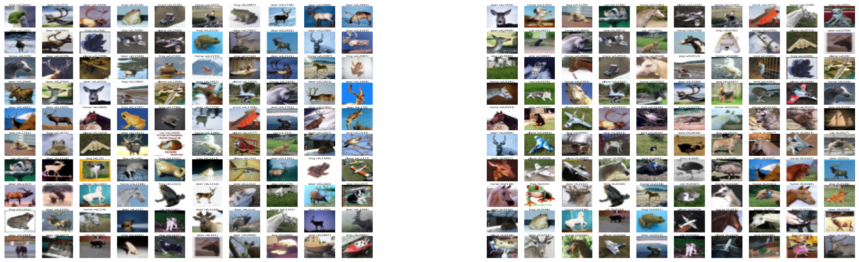
(g) Layer 17 vs layer 6.

(h) Layer 17 vs layer 7.

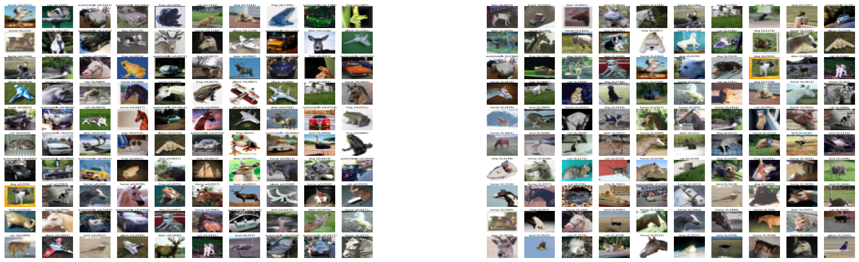
Figure 44: 100 most dissimilarly represented images comparing layers 0-9 with the penultimate layer (layer 17), *i.e.* 100 images that changed the most from one layer to the penultimate layer. The model was trained on CIFAR-10 training set with ResNet-18 architecture.



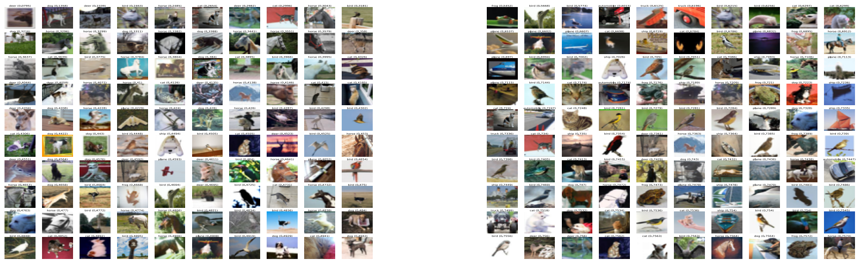
(a) Layer 17 vs layer 8. (b) Layer 17 vs layer 9.



(c) Layer 17 vs layer 10. (d) Layer 17 vs layer 11.



(e) Layer 17 vs layer 12. (f) Layer 17 vs layer 13.



(g) Layer 17 vs layer 14. (h) Layer 17 vs layer 15.

Figure 45: 100 most dissimilarly represented images comparing layers 9-17 with the penultimate layer (layer 17), *i.e.* 100 images that changed the most from one layer to the penultimate layer. The model was trained on CIFAR-10 training set with ResNet-18 architecture.

(a) Layer 17 vs layer 16.

(b)

Figure 46: 100 most dissimilarly represented images comparing layer 16 with the penultimate layer (layer 17), *i.e.* 100 images that changed the most from one layer to the penultimate layer. The model was trained on CIFAR-10 training set with ResNet-18 architecture.

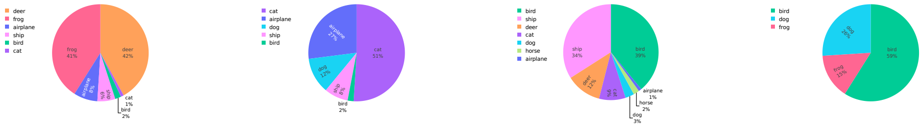
H USING POINTWISE ANALYSIS FOR NEURON INTERPRETABILITY

Pointwise representation similarity measures can be useful as a tool for model interpretability. In this Section, we showcase its use for interpreting the role of individual neurons in a representation. To understand what a specific neuron in a given layer is capturing, we can compare the representations of the full layer (with all Q neurons) to the representations of the layer without the neuron (with $Q - 1$ neurons). We can then observe how omitting the neuron affects the representation of each input in the test set, *i.e.* the lower the pontwise similarity score for an input i, the more its representation is altered by the removal of the neuron.

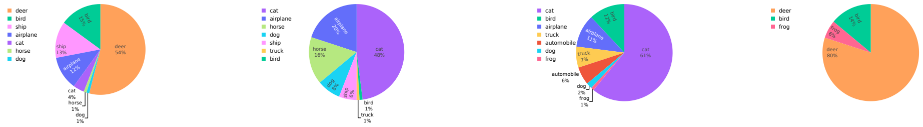
Thus, by inspecting and visualizing the least similar inputs when removing the neuron, one can gain insights into what patterns make a neuron's response unique from other neurons. We use this method to interpret what unique features the neurons at the penultimate layer of a ResNet-18 model (He et al., 2016) are capturing. We observe that some images obtain low similarity scores for some of the neurons removed and that most of those images pertain to one or two classes (shown in Figure 47). This indicates that the neurons in the penultimate layer are highly specialized in capturing features at the level of classes.

To validate the observation that many neurons primarily correspond to specific classes, for each class, we select $\approx$10% (50) of the neurons that have the highest ratio of images from that class in the 100 inputs that changed the most and train a linear probe on those. These neurons are the ones that best capture each class. Our hypothesis is that, if the 50 neurons are indeed capturing unique information about that class, the accuracy will increase significantly for that specific class. We also run the same experiment for the 50 neurons that least capture the corresponding classes as a baseline.

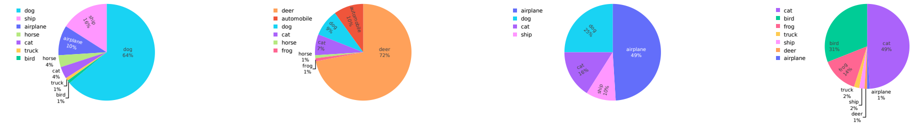
Table 7 shows that the models trained with the 50 most (least) informative neurons of a specific class achieve a higher (lower) accuracy for that class compared to the one that randomly selects 50 neurons. We additionally compare these results with other work in the literature (Cammarata et al., 2020; Bau et al., 2017), which analyzes the activations of neurons (*i.e.* the more a neuron activates, the more that neuron is excited by the features on those images), and show that both methods achieve similar results. However, while looking at activations show how much each neuron triggers for each input i, and can only be applied to this specific context, representation similarity measures inform what unique images, and features, each neuron is capturing, and can be generally applied to different contexts.



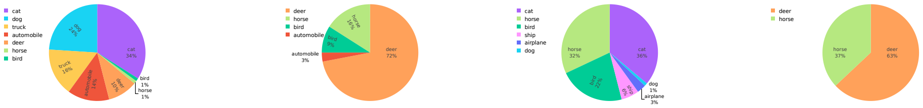
(a) Neuron 0. (b) Neuron 1. (c) Neuron 2. (d) Neuron 3.



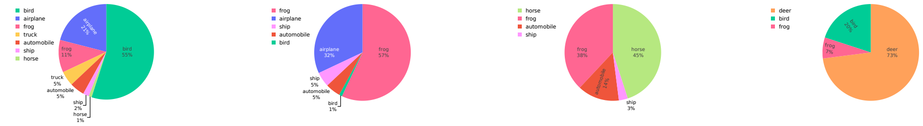
(e) Neuron 4. (f) Neuron 5. (g) Neuron 6. (h) Neuron 7.



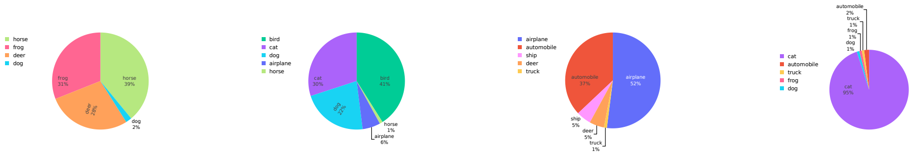
(i) Neuron 8. (j) Neuron 9. (k) Neuron 10. (l) Neuron 11.



(m) Neuron 12. (n) Neuron 13. (o) Neuron 14. (p) Neuron 15.



(q) Neuron 16. (r) Neuron 17. (s) Neuron 18. (t) Neuron 19.



(u) Neuron 20. (v) Neuron 21. (w) Neuron 22. (x) Neuron 23.

Figure 47: Class distribution of the 100 images with the lowest PNKA similarity, *i.e.* that changes the most its representation when removing the neuron, for the penultimate layer. We show that most of the neurons capture one or two class in its majority.

Table 7: Table with the results of the linear probes on 50 selected neurons. The linear probes trained on the 50 neurons that most (least) align with a specific class increase (decrease) its accuracy, compared to randomly picking 50 neurons. We also show that the results using PNKA achieves similar results with the results of using the neuron activation, which is a method specifically designed for interpretability of neurons. Due to size constraints, we altered the class names of Airplane and Automobile to Plane and Car, respectively.

		Overall	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Majority	#Neurons	512	49	21	53	105	91	41	38	66	20	28
>50%	#Neurons	278	25	15	28	49	55	27	16	34	12	17
	Random	0.930	0.963	0.959	0.878	0.767	0.960	0.952	0.941	0.939	0.960	0.965
Accuracy on the 50 *most* aligned neurons (PNKA)	*Plane*	0.770	**0.994**	0.955	0.810	0.345	0.828	0.366	0.857	0.745	0.906	0.923
	Car	0.720	0.864	**0.996**	0.774	0.510	0.694	0.805	0.848	0.000	0.879	0.824
	Bird	0.690	0.923	0.964	**0.994**	0.410	0.818	0.05	0.0	0.891	0.947	0.928
	Cat	0.830	0.914	0.921	0.839	**0.983**	0.748	0.635	0.641	0.846	0.796	0.960
	Deer	0.87	0.852	0.990	0.853	0.691	**0.997**	0.772	0.891	0.844	0.942	0.876
	Dog	0.800	0.911	0.793	0.808	0.654	0.737	**0.982**	0.95	0.305	0.923	0.975
	Frog	0.850	0.941	0.96	0.793	0.69	0.500	0.824	**0.990**	0.899	0.925	0.963
	Horse	0.610	0.94	0.953	0.816	0.174	0.011	0.308	0.006	**0.996**	0.934	0.923
	Ship	0.860	0.805	0.907	0.903	0.549	0.881	0.818	0.909	0.870	**0.991**	0.940
	Truck	0.750	0.810	0.882	0.380	0.584	0.325	0.831	0.898	0.832	0.956	**0.989**
Accuracy on the 50 *most* aligned neurons (activations)	*Plane*	0.8	**0.997**	0.723	0.826	0.664	0.721	0.742	0.816	0.914	0.758	0.863
	Car	0.77	0.814	**0.992**	0.805	0.333	0.62	0.689	0.859	0.696	0.967	0.885
	Bird	0.73	0.94	0.919	**0.991**	0.793	0.654	0.205	0.853	0.128	0.851	0.916
	Cat	0.72	0.726	0.526	0.683	**0.985**	0.672	0.394	0.643	0.715	0.868	0.953
	Deer	0.84	0.79	0.972	0.682	0.735	**0.997**	0.657	0.867	0.925	0.937	0.855
	Dog	0.68	0.9	0.925	0.822	0.023	0.821	**0.968**	0.007	0.414	0.906	0.97
	Frog	0.8	0.942	0.722	0.137	0.746	0.851	0.895	**0.993**	0.86	0.933	0.92
	Horse	0.73	0.925	0.748	0.688	0.807	0.648	0.516	0.057	**0.996**	0.924	0.946
	Ship	0.85	0.771	0.931	0.852	0.582	0.777	0.904	0.911	0.852	**0.995**	0.936
	Truck	0.62	0.852	0.885	0.865	0.339	0.683	0.626	0.0	0.921	0.069	**0.993**
Accuracy on the 50 *least* aligned neurons (PNKA)	*Plane*	0.88	**0.596**	0.98	0.928	0.587	0.975	0.95	0.942	0.943	0.956	0.963
	Car	0.88	0.952	**0.423**	0.9	0.873	0.828	0.928	0.964	0.964	0.961	0.977
	Bird	0.83	0.918	0.978	**0.018**	0.834	0.97	0.933	0.921	0.805	0.95	0.971
	Cat	0.85	0.982	0.987	0.886	**0.0**	0.948	0.921	0.95	0.963	0.948	0.898
	Deer	0.83	0.917	0.969	0.912	0.826	**0.0**	0.917	0.952	0.914	0.97	0.964
	Dog	0.87	0.881	0.962	0.934	0.788	0.968	**0.263**	0.955	0.958	0.964	0.978
	Frog	0.84	0.946	0.979	0.926	0.832	0.922	0.915	**0.0**	0.915	0.971	0.95
	Horse	0.83	0.918	0.934	0.921	0.818	0.844	0.938	0.973	**0.0**	0.969	0.974
	Ship	0.84	0.921	0.954	0.917	0.846	0.924	0.935	0.966	0.966	**0.0**	0.981
	Truck	0.85	0.972	0.959	0.877	0.794	0.928	0.95	0.97	0.94	0.945	**0.176**
Accuracy on the 50 *least* aligned neurons (activations)	*Plane*	0.91	**0.67**	0.977	0.903	0.87	0.979	0.896	0.95	0.937	0.982	0.894
	Car	0.87	0.966	**0.324**	0.921	0.871	0.896	0.925	0.96	0.925	0.94	0.981
	Bird	0.85	0.919	0.963	**0.0**	0.884	0.931	0.933	0.954	0.964	0.967	0.969
	Cat	0.84	0.964	0.983	0.917	**0.0**	0.97	0.782	0.954	0.944	0.969	0.936
	Deer	0.84	0.88	0.956	0.872	0.913	**0.0**	0.91	0.945	0.946	0.981	0.964
	Dog	0.87	0.907	0.965	0.951	0.831	0.93	**0.326**	0.968	0.959	0.946	0.966
	Frog	0.84	0.929	0.964	0.93	0.807	0.929	0.936	**0.0**	0.949	0.972	0.959
	Horse	0.83	0.887	0.934	0.879	0.859	0.892	0.928	0.975	**0.0**	0.977	0.976
	Ship	0.84	0.959	0.96	0.911	0.781	0.93	0.936	0.962	0.971	**0.02**	0.974
	Truck	0.86	0.959	0.968	0.867	0.856	0.936	0.935	0.976	0.874	0.965	**0.27**