Supplementary material for "Improving neural network representations using human similarity judgments"

Anonymous Author(s)
Affiliation
Address
email

A Experimental details

A.1 Model features

We extract penultimate layer features of four different ImageNet-models — AlexNet [7], VGG-16 [21], ResNet-18, and ResNet-50 [2] — and image encoder features of four different image/text models — CLIP RN50 and CLIP ViT-L/14 trained on WIT [16]; CLIP ViT-L/14 trained on Laion-400M [19] and Laion-2B [20] respectively. For extracting the model features, we use the Python library `thingsvision` [8].

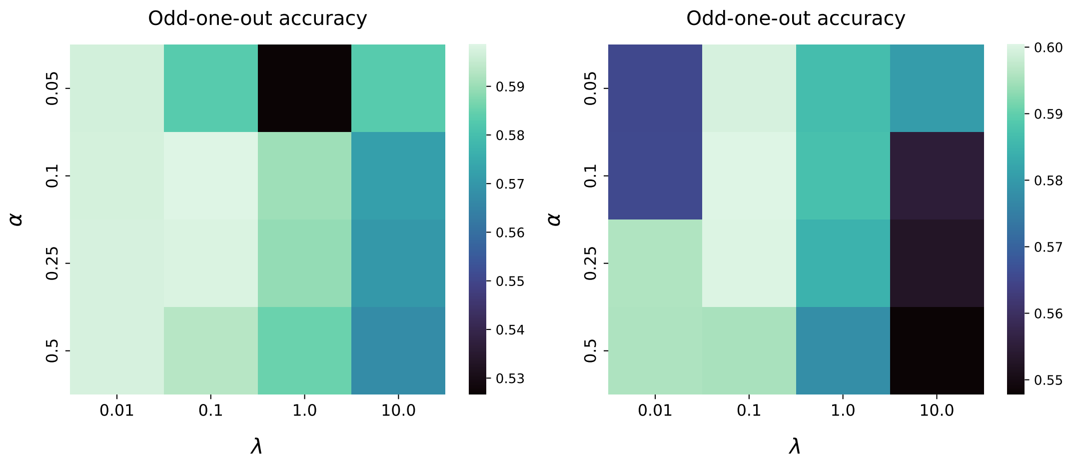
A.2 gLocal probing

To optimize the gLocal transforms, we use standard SGD with momentum and perform cross-validation according to the procedure proposed in Muttenthaler et al. [10]. For finding the optimal gLocal transform, we perform an extensive grid search over four different hyperparameter values — the learning rate, η, the strength of the regularization term λ, the global-local trade-off parameter α, and the temperature parameter, τ, used in the softmax expression for the local contrastive loss term (see Eq. 5). Specifically, we perform an extensive grid search over the Cartesian product of the following sets of hyperparameters:

- $\eta \in \{0.0001, 0.001, 0.01, 0.1\}$,
- $\lambda \in \{0.01, 0.1, 1.0, 10.0\}$,
- $\alpha \in \{0.05, 0.1, 0.25, 0.5, 1.0\}$,
- $\tau \in \{0.1, 0.25, 0.5, 1.0\}$.

We use the same η and λ grids for global probing. We use `PyTorch` [11] for implementing the probes and PyTorch lightning to accelerate training. We choose the gLocal transform that achieves the lowest alignment loss (see alignment term in Eq. 6). Among the values in the above grid, we find that a combination of $(\alpha = 0.1, \lambda = 0.1, \eta = 0.001)$ yields the lowest alignment loss/highest probing odd-one-out accuracy for both CLIP RN50 and CLIP ViT-L/14 (WIT) (see Figure A.1). A combination of $(\alpha = 0.25, \lambda = 0.1, \eta = 0.001)$ gives the second lowest alignment loss/highest probing odd-one-out accuracy for CLIP RN50 and CLIP ViT-L/14 (WIT).

For each (α, λ) combination we select that combination with the best probing odd-one-out accuracy on a held-out test among the set of possible learning rate, η, and temperature value, τ, combinations determined by the above grid. We observe that $\eta = 0.001$ generally gives the best results across the different (α, λ) combinations, whereas performance is fairly insensitive to the value of τ. Since neither $(\alpha = 0.1, \lambda = 0.1)$ nor $(\alpha = 0.25, \lambda = 0.1)$ are values at the edges of the hyperparameter



(a) CLIP RN50 (b) CLIP ViT-L/14 (WIT)

Figure A.1: Among all hyperparameter combinations considered in our grid search, a combination of ($\alpha = 0.1, \lambda = 0.1, \eta = 0.001$) for Eq. 6 in §3 yields the best odd-one-out accuracy on a held-out test set for both CLIP RN50 and CLIP ViT-L/14 (WIT).

grid, it is plausible to assume that both the contrastive local loss and the regularization term in Eq. 6 in §3 are necessary to obtain a transformation that leads to a *best-of-both-worlds* representation.

Although our goal has been to find a transform that induces both increased representational alignment and improved downstream task performance, we considered $\alpha = 1.0$ to examine whether downstream task performance can potentially be improved by excluding the alignment loss. Note that $\alpha = 1.0$ causes the optimization process to ignore the alignment loss. Unsurprisingly we did not find that to be the case. We remark that minimizing both the local contrastive loss and the regularization preserves the local similarity structure of the original representation space but does not inject any additional information into the representations. Moreover, it is non-trivial to choose a transform that works well across all downstream tasks without including the alignment loss. Therefore, we exclude $\alpha = 1.0$ in Figure A.1.

Compute. We used a compute time of approximately 400 hours on a single Nvidia A100 GPU with 40GB VRAM for all linear probing experiments — including the hyperparameter sweep. The computations were performed on a standard, large-scale academic SLURM cluster.

A.3 Few-shot learning

Here, we use n_s-fold cross-validation for finding the optimal ℓ_2-regularization parameter, where n_s refers to the number of shots per class. We select the parameter from the following set of values, $\{1\text{e}{+}6, 1, 1\text{e}{+}5, 1\text{e}{+}4, 1\text{e}{+}3, 1, 1\text{e}{+}2, 1\text{e}{+}1, 1, 1\text{e}{-}1, 1\text{e}{-}2, 1\text{e}{-}3, 1\text{e}{-}4\}$. We use the `scikit-learn` [12] implementation of (multinomial) logistic regression and refit the regression after selecting the optimal regularization parameter.

Compute. We used a compute time of approximately 5600 CPU-hours of 2.90GHz Intel Xeon Gold 6326 CPUs for all few-shot experiments. Computations were performed on a standard, large-scale academic SLURM cluster.

A.4 Anomaly Detection

In this section, we outline our anomaly detection experimental setting in more detail. In the anomaly detection settings that we consider in our analyses *normal*/*anomaly* classes are determined via the original classes in the data. Here, each of the original classes is once selected as a normal class with the remaining classes being anomalous and, vice versa, each class in the data is once selected as an anomalous class with the other classes being normal. After embedding the training images from either the normal or the anomalous class in a model's representation space, at inference time a model must classify whether a new image belongs to the normal data or whether it deviates from it and is thus considered an anomaly. For each example in the test set, a model yields an anomaly score where higher scores indicate more probability of an example being anomalous. Using the binary anomaly labels and the anomaly scores for each of the examples, we can then compute the *area-under-the-receiver-operating-characteristic-curve* (AUROC) to quantify the performance of the model.

One-vs-rest. Given a dataset (e.g., CIFAR-10) with C classes, one class (e.g., "airplane") is chosen to be the normal class and the remaining $C - 1$ classes of the dataset are considered anomalies. Each of the C classes is once selected as a normal class and the AUROC is averaged across the classes.

Leave-one-out (LOO). In contrast to the "one-vs-rest" setting, in LOO we define one class of the dataset as an anomaly and the remaining classes as normal. Similarly to the "one-vs-rest" setting, this results in C evaluations for a dataset with C classes.

In both "one-vs-rest" and LOO AD settings, we evaluate model representations in the following way: First, we compute the representations $\boldsymbol{X}_{\text{train}}$ of the normal samples in the train set. Then, we compute the representations of all test set examples $\boldsymbol{X}_{\text{test}}$. For each test set representation, we compute the cosine similarity to all normal train set representations, $\boldsymbol{X}_{\text{train}}$, and select the k nearest neighbor samples that have the highest cosine similarity.

The anomaly score of a test set representation is then defined as the average cosine distance to the k nearest train representations. k is a hyperparameter that determines the number of nearest neighbors over which the anomaly score is computed. We choose $k = 5$ for our experiments but show that performance is fairly insensitive to the value of k (see Tab. D.6).

Compute. For all AD experiments, we used a compute time of approximately 20 hours on a single Nvidia A100 GPU with 40GB VRAM. Computations were performed on a standard, large-scale academic SLURM cluster.

B What changes in the global structure of the representations after alignment?

In this section, we attempt to build some intuitions for how the global structure of the representations changes after alignment. To do so, we analyze the movements of the representations of items and superordinate categories in the THINGS dataset. Specifically, we compute cosine differences between the CLIP-ViT-L/14 representations of each pair of items in THINGS and then compute how these distances change under the transforms.

We show the pairs of items that change the most in distance in Table B.1. Items that are semantically related, like "curry" and "scrambled egg", tend to move closer together, and therefore have transformed distances that are smaller than their original distance. By contrast, items like "handcuff" and "stethoscope", which are semantically unrelated but perhaps have some slight visual similarity, tend to move farther apart. The distance changes under the gLocal transforms are correlated with, though generally less varied than, those under the naively-aligned transform.

To more broadly analyze the change in global structure, we then look at how distances between pairs of items change within and across superordinate categories (the top-down categories from THINGS). We show the results in Fig. B.1. Under the naively aligned transform, the items within each superordinate category tend to move slightly closer together — the diagonal is slightly blue — while the items from different categories tend to move substantially farther apart — the off-diagonal is mostly red. That is, the representations are broadly moving in a way that reflects the overall human semantic organization of the categories.

There are a few notable standouts: the categories of drink, food, plant, and animal change particularly much, and in particularly interesting ways. These categories each move much farther relative to all other categories (such as tool or musical instrument) than those other categories move relative to each other. This perhaps reflects the particular semantic salience of food, drink, plants, and animals from a human perspective. Furthermore, food and drink are one of the few pairs of superordinate categories between which distances actually decrease after the transform, presumably reflecting the strong semantic ties between these categories. Similarly, animals move less far from plants than from any other category, perhaps reflecting the fact that the animate/inanimate distinction is one of the strongest features in human semantic representations [18].

Under the gLocal transform, the pattern of changes is strongly correlated with the naively aligned transform ($r = 0.96$, $p \leq 10^{-16}$). However, in keeping with the regularization, the magnitude of the changes varies less.

Table B.1: Distances between pairs of individual items from THINGS, ranked by the relative change in cosine distance from before to after naive alignment (normalized by original distance). The top items move much closer together under naive alignment, while the bottom ones move much farther apart. (All results are from CLIP-ViT-L/14.)

Item 1	Item 2	original dist.	naively aligned dist.	gLocal dist.
curry	scrambled egg	0.303120	0.005276	0.401019
otter	warthog	0.305242	0.005530	0.382150
parfait	spaghetti	0.457553	0.009115	0.540346
otter	rhinoceros	0.327497	0.006530	0.456641
		⋮		
stethoscope	wheat	0.263908	1.284535	0.935891
grass	wallet	0.277866	1.347424	1.056572
cat	traffic light	0.285151	1.378671	0.981944
handcuff	sugar cube	0.272936	1.308337	0.904380

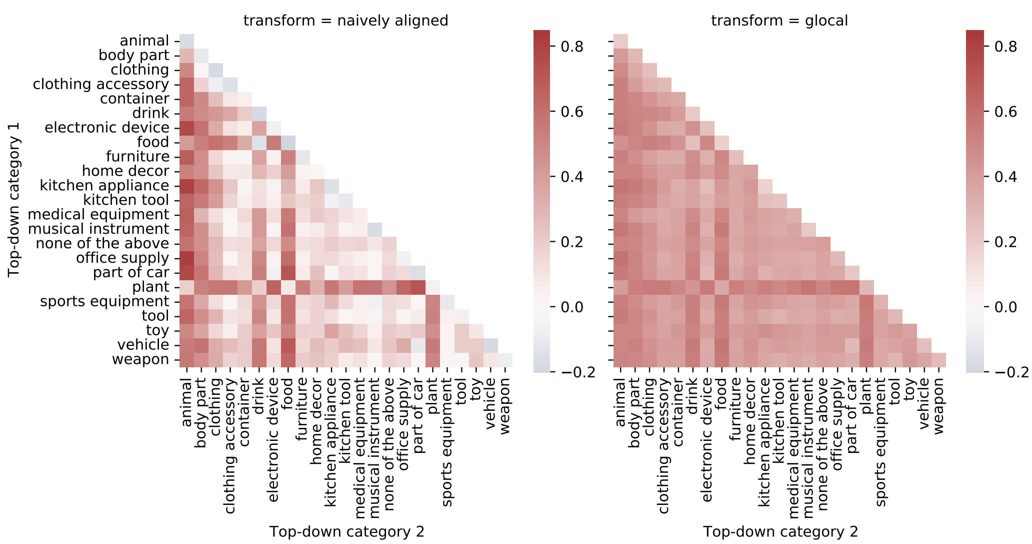


Figure B.1: How does the global structure of the representations change after alignment? Here, we analyze the movements of the representations of pairs of items from different superordinate categories from the THINGS dataset. The squares on the diagonal indicate the change in distance between items within a superordinate category, while the squares off the diagonal indicate changes between pairs of items from the corresponding pair of superordinate categories. A red color indicates the items from the categories move farther apart from each other after alignment, blue indicates moving closer together. Generally, items within a superordinate category move slightly closer together under naive alignment, while those in different categories move farther apart. A similar overall pattern is reflected in both the naively-aligned transform (left) and gLocal (right) ones, though under gLocal alignment there is a greater overall spreading of the representations. (All results are from CLIP-ViT-L/14.)

C Visualization of neighboring images

To provide further insight into the difference between the effects of the naive and gLocal transforms, in Figure C.1 we visualize the neighbors of nine anchor images. In order to show a diverse set of images, we pick the nearest neighbors in the CLIP ViT-L/14 (WIT) embedding space subject to the constraint that each neighbor comes from a different class from the original images and the nearer neighbors. In accordance with the results in §4.2, we find that the neighbors in the untransformed and gLocal spaces are generally similar, whereas neighbors in the naive representation space are frequently different. The naive transform appears to discard all non-semantic properties of images, whereas the untransformed and gLocal representation spaces are sensitive to pose, color, and numerosity. In cases where the closest neighbor differs between the naive and gLocal representations (third and fourth row), the neighbors in the gLocal representation are arguably better matches to the anchor.

Figure C.1: Comparison of neighbors in the ImageNet validation set for representations with different transforms. We visualize the 10 closest images subject to the constraint that each comes from a unique class. The anchor images are shown in the leftmost column. The three rows corresponding to each anchor image show their nearest neighbors in the untransformed, gLocal transformed, and naively transformed representations.

D Additional results on downstream tasks

In this section, we provide additional few-shot learning and anomaly detection results for all ImageNet and image/text models that we considered in our analyses (see §A.1). We start this section by demonstrating a strong relationship between the performances of the different downstream tasks.

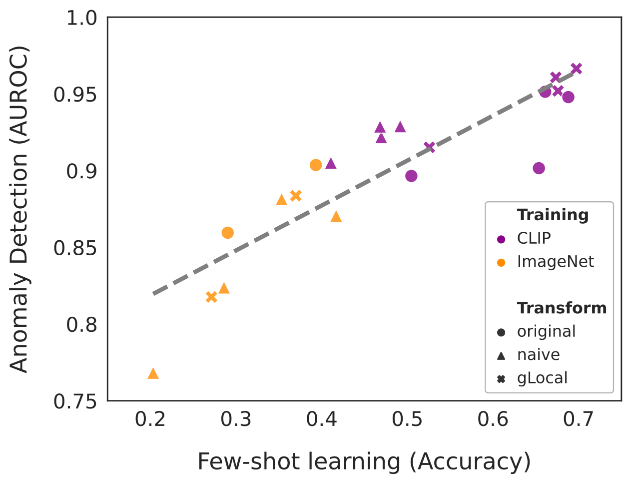


Figure D.1: Here, we show anomaly detection AUROC averaged across all tasks reported in Tab. {2, 3} as a function of the average 4-shot classification performance for all ImageNet and CLIP models (see §A.1), using either the original representations or the representations transformed via the naive or gLocal transformations.

Downstream task relationship. We observe a strong positive relationship ($r = 0.8872, p \leq 10^{-7}$) between the average few-shot learning and the average anomaly detection performance for all ImageNet and image/text models that we considered in our analyses (see Fig. D.1). This observation holds for both the original representation space and the representations transformed via the naive or gLocal transformations. This indicates that both downstream tasks require similar representations for similarly strong performance.

D.1 Few-shot learning

In the following section, we show additional few-shot learning results. Specifically, we report 4-shot performance of ImageNet models and show few-shot results as a function of the number of samples used during fitting.

Results for ImageNet models. In Tab. D.1 we report additional 4-shot results for ImageNet models. The gLocal transforms improve few-shot accuracy on Entity-{13,30} from BREEDS but the impact on few-shot performance is either inconsistent or negative for CIFAR-100 coarse, CIFAR-100, SUN397, and DTD of which the latter three are more fine-grained datasets than the other three.

Table D.1: 4-shot FS results using the original or transformed representations.

Model \ Transform	Entity-13		Entity-30		CIFAR100-Coarse		CIFAR100		SUN397		DTD	
	original	gLocal	original	gLocal	original	gLocal	original	gLocal	original	gLocal	original	gLocal
AlexNet	35.03	**39.59**	24.78	**25.85**	**30.51**	30.17	**26.26**	21.1	**24.19**	17.45	**33.39**	28.66
ResNet-18	56.15	**56.47**	**50.49**	50.03	38.3	**38.47**	**35.91**	34.42	**34.58**	33.17	**47.01**	44.28
ResNet-50	47.44	**51.41**	47.59	**50.22**	**48.2**	47.72	**45.29**	45.17	**44.69**	44.62	51.51	**51.85**
VGG-16	48.34	**54.76**	42.17	**44.04**	**36.74**	33.99	**31.77**	26.03	**34.55**	27.71	**42.35**	35.36

Effect of transforms for different numbers of training samples. When varying the number of training samples for the few-shot experiments described in §4.3 we observe consistent improvements of the gLocal transforms across shots. Excluding the high-variance setting of 2-shot learning, we either find stable improvements in accuracy for image/text models, or a downward trend for ImageNet models on some tasks. This corroborates our findings from §4.3. Results appear to be robust to changes in the training set size, in particular for the CLIP models. Yet, we observe the most substantial benefits in low data regimes. See Fig. D.2 for more details.

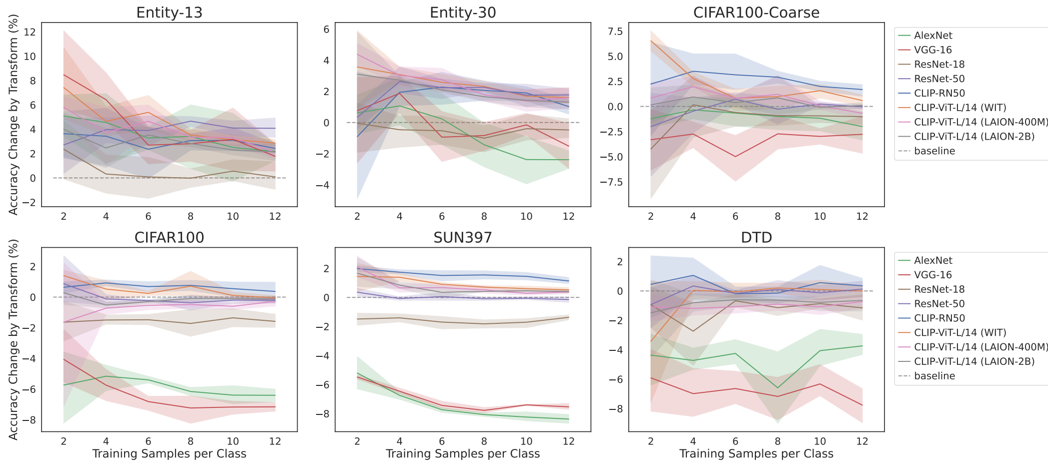


Figure D.2: Change in average accuracy for different numbers of training samples per super-class (top row) or (sub-)class (bottom row) used for few-shot learning. Error bands depict 95% Confidence Intervals (CIs), computed over 5 different runs.

D.2 Anomaly detection

In addition to the results of image/text models for the "one-vs-rest" anomaly detection (AD) setting that we presented in §4.4, here we show "one-vs-rest" AD performance of ImageNet models. While the gLocal transform considerably improves AD performance over the untransformed representations across the different datasets for image/text models (see Tab. {2, 3}, for ImageNet models we do not observe any improvements over the original representation space (see Tab. {D.2, D.3}).

Furthermore, we present results for the non-standard Leave-one-out (LOO) setting and for "CIFAR10-vs-CIFAR100" for all image/text and Imagenet models that we considered. In the "CIFAR10-vs-CIFAR100" AD task, all data of CIFAR10 is considered to be the normal class, and each sample from the CIFAR100 dataset is considered an anomaly. Similarly to the previously reported AD results, the gLocal transform substantially improves AD performance compared to the original representations for image/text models across all datasets but does not appear to have a considerable impact on the performance of ImageNet models (see Tab. {D.4, D.5}.

Table D.2: One-vs-rest nearest neighbor based AD results; with and without transformation. ImageNet30 results for ImageNet models are omitted due to overlap with train data.

	CIFAR10		CIFAR100		CIFAR100-Coarse		ImageNet30		DTD	
Model \ Transform	original	gLocal	original	gLocal	original	gLocal	original	gLocal	original	gLocal
AlexNet	**89.43**	85.63	**92.34**	88.53	**87.53**	82.75	_	_	**86.33**	79.51
ResNet-18	**92.19**	86.70	**95.06**	90.89	**92.16**	86.38	_	_	**94.38**	90.11
ResNet-50	**94.74**	94.13	**96.46**	96.18	**94.3**	94.03	_	_	**94.47**	94.42
VGG-16	**90.33**	88.00	**93.56**	91.97	**89.78**	88.16	_	_	**91.15**	85.5

Table D.3: One-vs-rest AD with a class distribution shift between train and test sets; with and without transformation.

	Entity-13		Entity-30		Living-17		Nonliving-26		Cifar100-shift	
Model \ Transform	original	gLocal	original	gLocal	original	gLocal	original	gLocal	original	gLocal
AlexNet	**83.84**	81.45	**85.38**	83.71	**87.04**	79.09	**81.45**	78.84	**80.21**	76.37
ResNet-18	**91.84**	89.45	**93.18**	91.6	**96.82**	93.1	**90.97**	89.87	**81.83**	77.44
ResNet-50	89.59	**91.26**	93.51	**93.86**	**98.27**	97.98	90.61	**91.85**	84.73	**85.38**
VGG-16	**89.78**	88.87	90.7	**91.56**	**94.72**	89.98	**89.78**	89.32	**83.42**	81.91

Table D.4: LOO nearest neighbor based AD results and "CIFAR-10 vs. CIFAR-100" AD results; with and without using the gLocal transform.

	CIFAR10		CIFAR100		Cifar100-Coarse		Cifar10 vs Cifar100	
Model \ Transform	original	gLocal	original	gLocal	original	gLocal	original	gLocal
AlexNet	**67.64**	62.7	**58.94**	55.83	**63.33**	59.37	**69.87**	68.01
ResNet-18	**72.35**	65.86	**64.86**	59.9	**71.38**	64.52	**81.42**	75.55
ResNet-50	**76.62**	75.36	**66.91**	66.03	**74.78**	73.96	**84.27**	84.17
VGG-16	**68.45**	64.31	**59.92**	57.89	**65.81**	64.28	73.55	**74.7**
CLIP-RN50	70.32	**72.46**	59.91	**61.43**	65.63	**68.07**	72.55	**76.78**
CLIP-ViT-L/14 (WIT)	84.91	**91.33**	67.08	**72.2**	73.48	**80.51**	85.24	**92.42**
CLIP-ViT-L/14 (LAION-400M)	**93.0**	92.37	74.05	**74.15**	82.13	**82.88**	94.44	**94.68**
CLIP-ViT-L/14 (LAION-2B)	93.55	**95.23**	76.88	**77.46**	84.67	**85.78**	93.18	**95.26**

Table D.5: LOO nearest neighbor based AD results; with and without using the gLocal transform.

	Entity-13		Entity-30		Living-17		Non-Living-26	
Model \ Transform	original	gLocal	original	gLocal	original	gLocal	original	gLocal
AlexNet	**62.05**	59.73	**58.2**	56.23	**61.02**	56.07	**56.27**	54.93
ResNet-18	**74.26**	68.88	**70.6**	64.7	**76.48**	70.79	**66.61**	63.82
ResNet-50	72.46	**73.67**	**73.37**	72.49	**83.5**	82.45	68.16	**68.29**
VGG-16	**70.26**	68.03	**66.38**	63.24	**73.31**	65.99	**64.87**	62.95
CLIP-RN50	69.99	**71.12**	63.49	**63.72**	**72.52**	70.04	62.55	**63.4**
CLIP-ViT-L/14 (WIT)	71.68	**74.88**	68.96	**70.58**	**82.9**	82.72	63.94	**67.33**
CLIP-ViT-L/14 (LAION-400M)	69.98	**71.44**	65.68	**66.26**	77.35	**77.4**	65.19	**66.27**
CLIP-ViT-L/14 (LAION-2B)	70.77	**72.49**	66.55	**67.68**	80.07	**80.09**	65.43	**67.99**

The nearest neighbor hyperparameter k. From the results reported in Tab. D.6 it can be inferred that the nearest neighbor hyperparameter k does not have a considerable impact on AD task performance across the different datasets. Here, we report the impact of k on the performance of CLIP ViT-L/14 (WIT) but the observation holds across all image/text models.

Table D.6: Nearest Neighbor AD performance of CLIP ViT-L/14 for different k.

k	2		5		10		20	
Dataset \ Transform	original	gLocal	original	gLocal	original	gLocal	original	gLocal
CIFAR-10	95.37	**98.16**	95.14	**98.16**	94.86	**98.11**	94.50	**98.04**
CIFAR-100	91.90	**97.08**	91.41	**97.04**	90.93	**96.92**	90.39	**96.75**
CIFAR-100-coarse	89.28	**95.68**	88.50	**95.59**	87.73	**95.4**	86.81	**95.12**
CIFAR-100-shift	74.48	**86.18**	73.69	**86.17**	73.00	**86.02**	72.29	**85.82**
ImageNet30	98.95	**99.78**	98.91	**99.79**	98.85	**99.8**	98.78	**99.8**
Entity-13	88.37	**92.89**	88.54	**93.57**	88.45	**93.94**	88.28	**94.22**
Entity-30	91.26	**95.36**	91.31	**95.77**	91.22	**95.97**	91.03	**96.12**

D.3 Global versus gLocal transform

Aside from the AD performance of CLIP RN50 and CLIP ViT-L/14 (WIT), the gLocal transform leads to more substantial improvements on downstream tasks than the global transform. In Tab D.7, we report the average few-shot and anomaly detection performances using the global or gLocal transforms. For FS, we average performance over all results reported in Tab. 1, and for AD we average performance across all results reported in Tab. {2, 3, D.1}.

E Representational alignment

E.1 Human similarity judgments and RSMs

Multi-arrangement task. Human similarity judgments for King et al. [5] and [1] were obtained by using a multi-arrangement task. In a multi-arrangement task, participants are presented with a computer screen showing images of a number of different objects. The participants are asked to arrange the images into semantically meaningful clusters, given the instruction that images of objects that lie close together are considered more similar. From this arrangement, one can infer pairwise (dis-)similarities of the objects and average those across all participants to obtain a representative (dis-)similarity matrix.

Table D.7: Comparison of the average downstream task performance global and gLocal transforms.

Model \ Transform	AD global	AD gLocal	FS global	FS gLocal
AlexNet	81.16	**81.76**	26.65	**27.14**
ResNet-18	84.62	**88.39**	40.75	**42.80**
ResNet-50	93.19	**93.23**	48.29	**48.50**
VGG-16	87.32	**88.36**	35.36	**36.98**
CLIP-RN50	**92.12**	91.52	50.02	**52.57**
CLIP-ViT-L/14 (WIT)	**95.49**	95.14	65.80	**67.44**
CLIP-ViT-L/14 (LAION-400M)	95.72	**96.08**	66.82	**67.34**
CLIP-ViT-L/14 (LAION-2B)	96.33	**96.65**	69.73	**69.74**

Ordinal scale. In Peterson et al. [13, 14], pairwise similarity judgments were obtained by asking human participants to rate the similarity of pairs of objects on an ordinal scale that ranges from 0 ("not similar at all") to 10 ("very similar"). The pairwise similarity ratings can be averaged across the different participants which in turn yields a matrix of similarities between pairs of objects.

Triplet odd-one-out choices. The triplet odd-one-out task is a commonly used task in the cognitive sciences to infer pairwise object similarity ratings [17, 3, 9]. The triplet odd-one-out task is a *three-alternative-forced-choice* task where participants are presented with three objects and have to select the one that does not fit. In contrast to the multi-arrangement task or an ordinal scale, the triplet odd-one-out task does not naturally yield a similarity matrix. A similarity matrix can be obtained, however, by learning representations for the objects being used in the task from the human responses. Variational Interpretable Concept Embeddings (VICE) — an approximate Bayesian method for inferring mental representations of object concepts from triplet odd-one-out choices — is a method that was specifically developed for that purpose. VICE uses variational inference to learn representations for the objects in the triplets by fitting the human responses via stochastic gradient descent. The method minimizes $\mathcal{L}_{\text{global}}$ with additional non-negativity and sparsity constraints on the representations. More details about the optimization can be found in Muttenthaler et al. [9]. From the VICE solution, one can easily compute a representational similarity matrix (RSM). Specifically, given learned object representations $\boldsymbol{V} \in \mathbb{R}^{n \times d}$, one first computes the dot-product similarity matrix $\boldsymbol{S}_h := \boldsymbol{V}\boldsymbol{V}^\top$ and then exponentiate this matrix elementwise, $\boldsymbol{S}'_h := \exp(\boldsymbol{S}_h)$. One can then apply the softmax function defined in Eq. 1 to every combination of triplets in the exponentiated similarity matrix which yields the final RSM for triplet odd-one-out choices from Hebart et al. [4]. The last step is performed to guarantee that the pairwise similarities are modeled according to the triplet odd-one-out objective function that was used to learn the human object representations $\boldsymbol{V}$ (see Eq. 2).

E.2 Neural network representations and RSMs

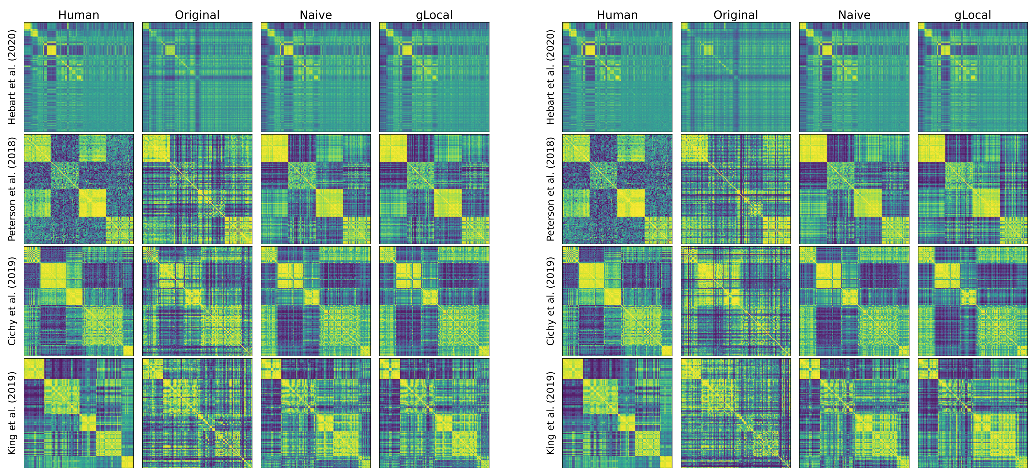
Neural network representations. RSMs for neural network representations are obtained by first embedding the same set of images that were presented to the human participants in the p-dimensional latent space of a neural net. The latent space could be any layer of a neural network. Here we use the penultimate layer space for ImageNet models and the image encoder space for image/text models. We do this because previous work has shown that the penultimate layer space of ImageNet models and the image encoder space of image/text models respectively yield the highest similarity to human behavior [14, 15, 10]. After embedding the images into the neural net's latent space, one obtains a representation matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ for the n images in the data. Instead of simply computing the dot-product similarity matrix $\boldsymbol{S} := \boldsymbol{X}\boldsymbol{X}^\top$, in RSA one typically uses either a cosine similarity or a Pearson correlation kernel to compute the affinity matrix,

$$\cos(\boldsymbol{x}_i, \boldsymbol{x}_j) := \frac{\boldsymbol{x}_i^\top \boldsymbol{x}_j}{||\boldsymbol{x}_i||_2 ||\boldsymbol{x}_j||_2}; \qquad \phi(\boldsymbol{x}_i, \boldsymbol{x}_j) := \frac{(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^\top (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_j)}{||(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)||_2 ||(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_j)||_2},$$

where the cosine similarity kernel function $\cos(\boldsymbol{x}_i, \boldsymbol{x}_j)$ or the Pearson correlation kernel function $\phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is applied to every $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ vector pair of the matrix $\boldsymbol{X}$ for obtaining the final representational similarity matrix $\boldsymbol{S}' \in \mathbb{R}^{n \times n}$. Here, we use the Pearson correlation kernel function $\phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to obtain a neural net's RSM. Pearson correlation is the centered version of cosine similarity and the ranking of the obtained similarities does not differ between the two kernel functions but Pearson correlation first centers the vectors to have zero mean and is therefore a more robust measure. For obtaining RSMs with transformed representations, the transforms are first applied to $\boldsymbol{X}$ before computing $\boldsymbol{S}'$.

E.3 Representational Similarity Analysis (RSA)

Additional RSMs. To corroborate our findings from §4.5, here we additionally show RSMs for CLIP RN50 and CLIP ViT-L/14 (Laion 2B). In accordance with the different RSMs obtained from the representation space of CLIP ViT-L/14 (WIT), there does not appear to be a qualitative difference in the global similarity structure between the RSMs obtained from applying either the naive or the gLocal transforms to CLIP RN50 or CLIP ViT-L/14 (Laion 2B) (see Fig. E.1). Hence, the gLocal transform improves representational alignment while preserving the local similarity structure of the original representation equally well for the different CLIP models, as we show in Tab. 4.



(a) CLIP RN50

(b) CLIP ViT-L/14 (Laion 2B)

Figure E.1: Here, we show representational similarity matrices (RSMs) for human behavior and CLIP RN50 [WIT; 16] and CLIP ViT-L/14 [Laion 2B; 20] for four different human similarity judgment datasets [3, 14, 1, 5]. We contrast RSMs obtained from the network's original representation space (second column), the naively transformed representation space [10] (third column), and the representation space obtained by using the gLocal transform (rightmost column) against RSMs directly constructed from human similarity judgments (leftmost column).

F Global transform derivation

Here we derive that

$$\min_{\alpha} \|\boldsymbol{W} - \alpha I\|_{\mathrm{F}}^2 = \|\boldsymbol{W} - \left(\textstyle\sum_{i=1}^{p} \boldsymbol{W}_{ii}/p\right) I\|_{\mathrm{F}}^2 .$$

First, observe that

$$\begin{aligned}\min_{\alpha} \|\boldsymbol{W} - \alpha I\|_{\mathrm{F}}^2 &= \min_{\alpha} \sum_{i=1}^{p} \sum_{j=1}^{p} \left(\boldsymbol{W}_{ij} - \alpha \mathbb{1}_{[i=j]}\right)^2 \\ &= \min_{\alpha} \sum_{i=1}^{p} \sum_{j=1, j \neq i}^{p} \boldsymbol{W}_{ij}^2 + \sum_{k=1}^{p} \left(\boldsymbol{W}_{kk} - \alpha\right)^2 \\ &= \sum_{i=1}^{p} \sum_{j=1, j \neq i}^{p} \boldsymbol{W}_{ij}^2 + \min_{\alpha} \sum_{k=1}^{p} \left(\boldsymbol{W}_{kk} - \alpha\right)^2 .\end{aligned}$$

The minimizer of $\min_\alpha \sum_{k=1}^p (\boldsymbol{W}_{kk} - \alpha)^2$ is attained with $\alpha = \sum_{\ell=1}^p \boldsymbol{W}_{\ell\ell}/p$. Substituting this back into the last equality and reversing the steps from before we have

$$\begin{aligned}
\sum_{i=1}^p \sum_{j=1, j\neq i}^p \boldsymbol{W}_{ij}^2 + \min_\alpha \sum_{k=1}^p (\boldsymbol{W}_{kk} - \alpha)^2 &= \sum_{i=1}^p \sum_{j=1, j\neq i}^p \boldsymbol{W}_{ij}^2 + \sum_{k=1}^p \left(\boldsymbol{W}_{kk} - \sum_{\ell=1}^p \boldsymbol{W}_{\ell\ell}/p \right)^2 \\
&= \sum_{i=1}^p \sum_{j=1}^p \left(\boldsymbol{W}_{ij} - \left(\sum_{\ell=1}^p \boldsymbol{W}_{\ell\ell}/p \right) \mathbb{1}_{[i=j]} \right)^2 \\
&= \| \boldsymbol{W} - (\textstyle\sum_{\ell=1}^p \boldsymbol{W}_{\ell\ell}/p) \, I \|_{\mathrm{F}}^2 ,
\end{aligned}$$

which finishes our derivation.

G Properties of LCKA

Kornblith et al. [6] previously validated linear centered kernel alignment (LCKA) as a way to measure similarity between neural network representations. Given representations $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{Y} \in \mathbb{R}^{n\times p_2}$ containing embeddings of the same n images stacked row-wise, LCKA is:

$$\mathrm{LCKA}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\langle \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top, \tilde{\boldsymbol{Y}}\tilde{\boldsymbol{Y}}^\top \rangle_{\mathrm{F}}}{\|\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top\|_{\mathrm{F}} \|\tilde{\boldsymbol{Y}}\tilde{\boldsymbol{Y}}^\top\|_{\mathrm{F}}} = \frac{\|\tilde{\boldsymbol{X}}^\top \tilde{\boldsymbol{Y}}\|_{\mathrm{F}}^2}{\|\tilde{\boldsymbol{X}}^\top\tilde{\boldsymbol{X}}\|_{\mathrm{F}} \|\tilde{\boldsymbol{Y}}^\top\tilde{\boldsymbol{Y}}\|_{\mathrm{F}}}, \tag{1}$$

where $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$ are equal to $\boldsymbol{X}$ and $\boldsymbol{Y}$ with column means subtracted. (Formally, $\tilde{\boldsymbol{X}} = \boldsymbol{H}\boldsymbol{X}$ and $\tilde{\boldsymbol{Y}} = \boldsymbol{H}\boldsymbol{Y}$ and $\boldsymbol{H} = \boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is the centering matrix, which is a matrix representation of the linear operator that subtracts column means.)

As Kornblith et al. [6] note, linear CKA can be thought of as measuring the cosine similarity between all pairs of principal components (PCs) of $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$, weighted by the products of the proportions of variance these PCs explain in each representation. Formally, let $\tilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ and $\tilde{\boldsymbol{Y}} = \boldsymbol{U}'\boldsymbol{\Sigma}'\boldsymbol{V}'^\top$ be the singular value decompositions of $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$. The left-singular vectors $\boldsymbol{u}_i = \boldsymbol{U}_{:,i}$ are the (unit-norm) PCs of $\boldsymbol{X}$, and the squared singular values $\lambda_i = \Sigma_{ii}^2$ are the amount of variance that those PCs explain (up to a factor of $1/n$). Given these singular value decompositions, linear CKA is:

$$\mathrm{LCKA}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \lambda_i \lambda'_j \left(\boldsymbol{u}_i^\top \boldsymbol{u}'_j \right)^2}{\sqrt{\sum_{i=1}^{p_1} \lambda_i^2} \sqrt{\sum_{j=1}^{p_2} {\lambda'_j}^2}}. \tag{2}$$

References

[1] Radoslaw M. Cichy, Nikolaus Kriegeskorte, Kamila M. Jozwik, Jasper J.F. van den Bosch, and Ian Charest. The spatiotemporal neural dynamics underlying perceived similarity for real-world objects. *NeuroImage*, 194:12–24, 2019. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2019.03.031.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] Martin N. Hebart, Charles Y. Zheng, Francisco Pereira, and Chris I. Baker. Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. *Nature Human Behaviour*, 4(11):1173–1185, October 2020. doi: 10.1038/s41562-020-00951-3.

[4] Martin N Hebart, Oliver Contier, Lina Teichmann, Adam H Rockter, Charles Y Zheng, Alexis Kidder, Anna Corriveau, Maryam Vaziri-Pashkam, and Chris I Baker. Things-data, a multimodal collection of large-scale datasets for investigating object representations in human brain and behavior. *eLife*, 12:e82580, feb 2023. ISSN 2050-084X. doi: 10.7554/eLife.82580.

[5] Marcie L. King, Iris I.A. Groen, Adam Steel, Dwight J. Kravitz, and Chris I. Baker. Similarity judgments and cortical visual responses reflect different properties of object and scene categories in naturalistic images. *NeuroImage*, 197:368–382, 2019. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2019.04.079.

[6] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

[7] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv e-prints*, art. arXiv:1404.5997, April 2014.

[8] Lukas Muttenthaler and Martin N. Hebart. Thingsvision: A python toolbox for streamlining the extraction of activations from deep neural networks. *Frontiers in Neuroinformatics*, 15:45, 2021. ISSN 1662-5196. doi: 10.3389/fninf.2021.679838.

[9] Lukas Muttenthaler, Charles Y Zheng, Patrick McClure, Robert A Vandermeulen, Martin N Hebart, and Francisco Pereira. VICE: Variational Interpretable Concept Embeddings. *Advances in Neural Information Processing Systems*, 35:33661–33675, 2022.

[10] Lukas Muttenthaler, Jonas Dippel, Lorenz Linhardt, Robert A Vandermeulen, and Simon Kornblith. Human alignment of neural network representations. In *11th International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, Mai 01-05, 2023*. OpenReview.net, 2023.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019.

[12] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[13] Joshua C. Peterson, Joshua T. Abbott, and Thomas L. Griffiths. Adapting deep network features to capture psychological representations. In Anna Papafragou, Daniel Grodner, Daniel Mirman, and John C. Trueswell (eds.), *Proceedings of the 38th Annual Meeting of the Cognitive Science Society, Recogbizing and Representing Events, CogSci 2016, Philadelphia, PA, USA, August 10-13, 2016*. cognitivesciencesociety.org, 2016.

[14] Joshua C. Peterson, Joshua T. Abbott, and Thomas L. Griffiths. Evaluating (and improving) the correspondence between Deep Neural Networks and Human Representations. *Cogn. Sci.*, 42(8): 2648–2669, 2018. doi: 10.1111/cogs.12670.

[15] Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 9616–9625. IEEE, 2019. doi: 10.1109/ICCV.2019.00971.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021.

[17] Rocco Robilotto and Qasim Zaidi. Limits of lightness identification for real objects under natural viewing conditions. *Journal of Vision*, 4(9):9–9, 09 2004. ISSN 1534-7362. doi: 10.1167/4.9.9.

[18] Timothy T Rogers, James L McClelland, et al. *Semantic cognition: A parallel distributed processing approach*. MIT press, 2004.

[19] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: open dataset of clip-filtered 400 million image-text pairs. *CoRR*, abs/2111.02114, 2021.

[20] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.