The pitfalls of measuring representational similarity using representational similarity analysis

Marin Dujmović[1*], Jeffrey S Bowers[1], Federico Adolfi[1,2], and Gaurav Malhotra[1]

[1] *School of Psychological Science, University of Bristol, Bristol, UK*

[2] *Ernst-Strüngmann Institute for Neuroscience in Cooperation with Max-Planck Society, Frankfurt, Germany*

[*] *marin.dujmovic@bristol.ac.uk*

April 1, 2022

Abstract

A core challenge in neuroscience is to assess whether diverse systems represent the world similarly. Representational Similarity Analysis (RSA) is an innovative approach to address this problem and has become increasingly popular across disciplines from machine learning to computational neuroscience. Despite these successes, RSA regularly uncovers difficult-to-reconcile and contradictory findings. Here we demonstrate the pitfalls of using RSA to infer representational similarity and explain how contradictory findings arise and support false inferences when left unchecked. By comparing neural representations in primate, human and computational models, we reveal two problematic phenomena that are ubiquitous in current research: a "mimic" effect, where confounds in stimuli can lead to high RSA scores between provably dissimilar systems, and a "modulation effect", where RSA-scores become dependent on stimuli used for testing. Since our results bear on existing findings and inferences, we provide recommendations to avoid these pitfalls and sketch a way forward.

Introduction

How do other animals see the world? Do different species represent the world in a similar manner? How do the internal representations of AI systems compare with humans and animals? The traditional scientific method of probing internal representations of humans and animals (popular in both psychology and neuroscience) relates them to properties of the external world. By moving a line across the visual field of a cat, [1] found out that neurons in the visual cortex represent edges moving in specific directions. In another Nobel-prize winning work, [2] discovered that neurons in the hippocampus represent the location of an animal in the external world. Despite these successes it has proved difficult to relate internal representations to more complex properties of the world. Moreover, relating representations across individuals and species is challenging due to the differences in experience across individuals and differences of neural architectures across species.

These challenges have led to recent excitement around Representation Similarity Analysis (RSA) which appears to overcome many of these obstacles. RSA usually takes patterns of activity from two systems and computes how the distances between activations in one system correlate with the distances between corresponding activations in the second system (see Figure 1). Rather than compare each pattern of activation in the first system directly to the corresponding pattern of activation in the second system, it computes a second-order measure of similarity, comparing the systems based on their *representational geometries*. The advantage of looking at representational geometries is that one no longer needs to match the architecture of two systems, or even the format of the initial activity patterns (see Supplementary Information, Section A for a brief history of RSA and its philosophical origins). One could compare, for example, fMRI signals with single cell recordings, EEG traces with behavioural data, or vectors in a computer algorithm with

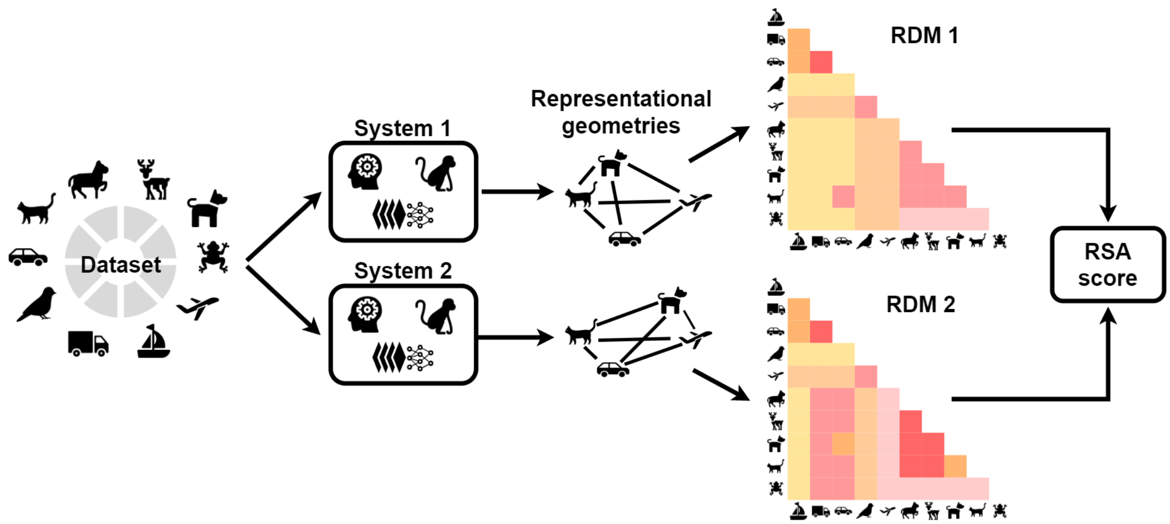

Figure 1: **RSA calculation.** A series of stimuli from a set of categories (or conditions) are used as inputs to two different systems (for example, a human brain and a primate brain). Activity from regions of interest is recorded for each stimulus. Pair-wise distances in activity patterns are calculated to get the representational geometry of each system. This representational geometry is expressed as a representational dissimilarity matrix (RDM) for each system. Finally, an RSA score is determined by computing the correlation between the two RDMs.

spiking activity of neurons [3]. RSA is now ubiquitous in computational psychology and neuroscience and has been applied to compare object representations in humans and primates [4], representations of visual scenes by different individuals [5, 6], representations of visual scenes in different parts of the brain [7], to study specific processes such as cognitive control [8] or the dynamics of object processing [9], and most recently, to relate neuronal activations in human (and primate) visual cortex with activations of units in Deep Neural Networks [10–14].

However, some recent research suggests that RSA may be an unreliable measure of how similarly two systems represent the world. For example, many studies [15–19] have shown that Convolutional Neural Networks (CNNs), trained on standard image datasets, such as `ImageNet`, classify input images based on shortcuts, such as their texture. Activations in these same networks also show a high RSA with activations in the human and primate inferior temporal cortex [10, 11], even though it is well-known that humans primarily represent objects based on their global properties such as shape, rather than shortcuts, such as texture [20–22]. Similarly, some studies using RSA have shown that the hierarchy of representations in the ventral visual stream in humans and primates correlates with the hierarchy of representations in the layers of a CNN – i.e., deeper layer in a CNN have a higher RSA with deeper layer in the visual ventral stream [10]. But [23] have recently shown that this correspondence is dataset-dependent and does not replicate for some naturalistic and artificial stimuli.

How is it possible for two systems to have a high RSA score but represent different features of inputs? Through a series of simulations that capture increasingly plausible training and testing scenarios, we demonstrate the properties of datasets and procedures that, in practice, lead to high RSA scores between mechanistically dissimilar systems. The experiments showcasing these pitfalls span the entire spectrum from artificial intelligence to computational neuroscience, involving comparisons within and between sets of artificial and biological systems. In particular, we shed light on two problematic phenomena that bear on any efforts to compare systems based on RSA: 1) the presence of confounds in the training data which leads systems to mimic each other's representational geometry even in the absence of mechanistic similarity, 2) the artifactual modulation of RSA scores due to the intrinsic structure of datasets rather than system alignment. Our demonstrations provide an explanation of how these phenomena, which arise ubiquitously, underlie contradictory and paradoxical findings in the literature. Since our results have considerable generality with respect to current practices across multiple fields, we discuss the implications for published results and prevailing interpretations, and provide broadly applicable recommendations to move forward.

Results

Simulation 1: RSA between systems with different transformations

We will now show an example of how two systems can end up with very similar representational geometries even though they (i) select different features of inputs and (ii) transform their inputs through very different functions. Consider a set of stimuli, $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ from two classes that form two clusters in the input space as shown in Figure 2A. Let us assume that each stimulus, $\boldsymbol{x_i}$ contains multiple features that independently predict the class of the stimulus. We will call each of these predictive features *confounds*. For example, shape and texture can be confounds when classifying an image as belonging to `DOG` or `AEROPLANE` classes if each feature can be independently used to predict whether an image belongs to the `DOG` or `AEROPLANE` class. Consider two recognition systems Φ_1 and Φ_2 that map each input stimulus, $\boldsymbol{x_i}$, to an internal representation using their respective transformation functions, $\Phi_1(\boldsymbol{x_i})$ and $\Phi_2(\boldsymbol{x_i})$. Furthermore, we will assume that Φ_1 and Φ_2 are qualitatively different functions and act on different features of the input. We are interested in showing that such qualitatively different functions acting on different features can nevertheless end up with similar representational geometries.

The representational distance, $d[\boldsymbol{x_i}, \boldsymbol{x_j}]$, between the projections of any pair of input stimuli, $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, is proportional to the inner product between their projection in the

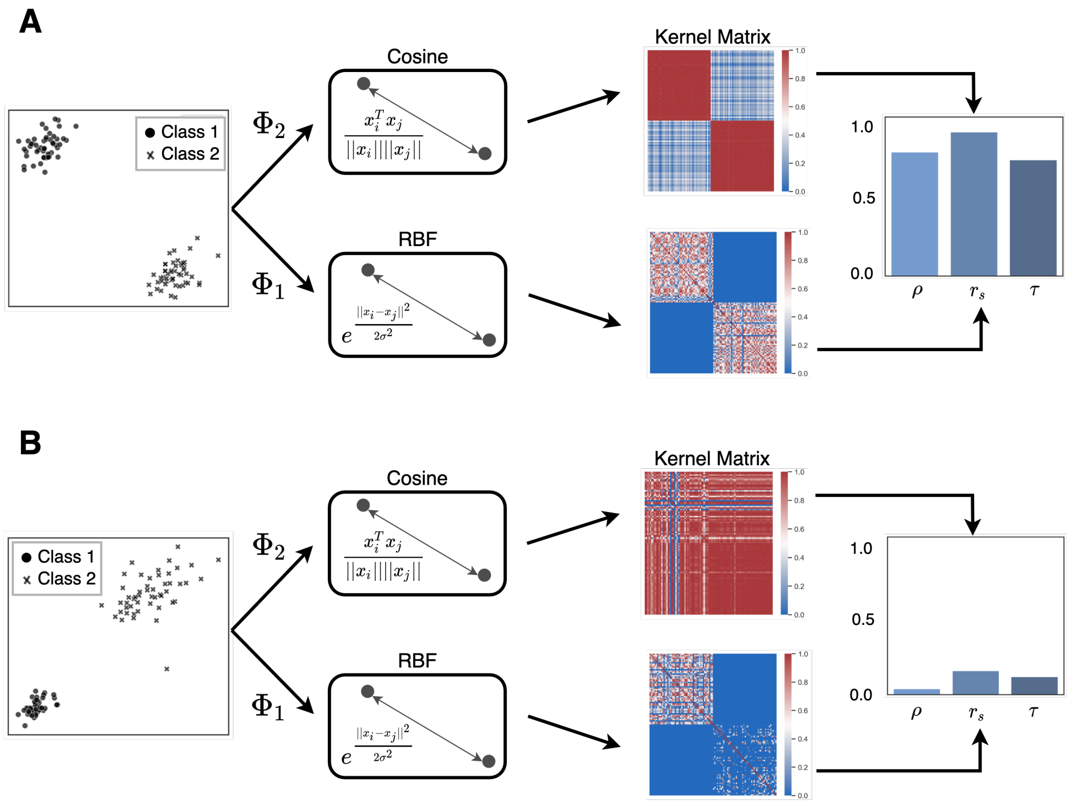

Figure 2: **RSA between two systems with known transformations.** In each panel a set of 2D stimuli are transformed using two different functions (Φ_1 and Φ_2), which project these stimuli into two different representational spaces. The distance between these projections are given by the RBF and Cosine kernels, respectively (see main text). The geometry of these projections can be visualised using the kernel matrices, which show the pair-wise distances between all stimuli in the representational space. The bar graph on the right-hand-side shows the RSA-score computed as a Pearson correlation (ρ), Spearman's rank correlation (r_s) and Kendall's rank correlation (τ). We can see that the input stimuli in Panel A leads to a high correlation in the representational geometry of the two systems, while the input stimuli in Panel B leads to a low correlation, even though the transformations remain the same.

feature space:

$$d[\boldsymbol{x_i}, \boldsymbol{x_j}] \propto \Phi(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{x_j}) \tag{1}$$

Thus, we can obtain the representational geometry of the input stimuli, $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$, by computing the pairwise distances, $d[\boldsymbol{x_i}, \boldsymbol{x_j}]$ for all pairs of data points, (i, j). Here, we assume that the projections Φ_1 and Φ_2 are such that these pairwise distances are given by two positive semi-definite kernel functions $\kappa_1(\boldsymbol{x_i}, \boldsymbol{x_j})$ and $\kappa_2(\boldsymbol{x_i}, \boldsymbol{x_j})$, respectively:

$$\kappa_1(\boldsymbol{x_i}, \boldsymbol{x_j}) = \Phi_1(\boldsymbol{x_i}) \cdot \Phi_1(\boldsymbol{x_j}) \tag{2}$$

$$\kappa_2(\boldsymbol{x_i}, \boldsymbol{x_j}) = \Phi_2(\boldsymbol{x_i}) \cdot \Phi_2(\boldsymbol{x_j}) \tag{3}$$

Now, let us consider two qualitatively different kernel functions: $\kappa_1(\boldsymbol{x_i}, \boldsymbol{x_j}) = e^{\frac{||\boldsymbol{x_i} - \boldsymbol{x_j}||^2}{2\sigma^2}}$ is a radial-basis kernel (where σ^2 is the bandwidth parameter of the kernel), while $\kappa_2(\boldsymbol{x_i}, \boldsymbol{x_j}) = \frac{\boldsymbol{x_i^T x_j}}{||\boldsymbol{x_i}||||\boldsymbol{x_j}||}$ is a cosine kernel. Figure 2A shows a dataset of points in a 2D input space that are projected by two different systems into a cosine and RBF kernel space. Since the cosine and RBF kernels are Mercer kernels [24, 25], each kernel matrix in Figure 2A shows the pairwise distances (as measured by the inner product) between data points projected in the two feature spaces. We can determine how the geometry of these projections in the two systems relate to each other by computing the correlation between the kernel matrices, shown on the right-hand-side of Figure 2A. We can see from these results that the kernel matrices are highly correlated – i.e., the input stimuli are projected to very similar geometries in the two representational spaces.

If one did not know the input transformations and simply observed the correlation between kernel matrices, it would be tempting to infer that the two systems Φ_1 and Φ_2

transform an unknown input stimulus $\boldsymbol{x}$ through a similar set of functions – for example functions that belong to the same class or project inputs to similar representational spaces. However, this would be an error. The projections $\Phi_1(\boldsymbol{x})$ and $\Phi_2(\boldsymbol{x})$ are fundamentally different – Φ_1 (radial basis kernel) projects an input vector into an infinite dimensional space, while Φ_2 (cosine kernel) projects it onto a unit sphere. The difference between these functions becomes apparent if one considers how this correlation changes if one considers a different set of input stimuli. For example, the set of data points shown at the left of Figure 2B, are projected to very different geometries, leading to a low correlation between the two kernel matrices (right-hand side).

In fact, the reason for highly correlated kernel matrices in Figure 2A is not a similarity in the transformations Φ_1 and Φ_2 but the structure of the dataset. The representational distance between any two points $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ in Φ_1 is a function of their Euclidean distance $||\boldsymbol{x_i} - \boldsymbol{x_j}||$, while in Φ_2, it is a function of their cosine distance, $\boldsymbol{x_i^T x_j}$. These two features – Euclidean distance and cosine distance – mimic each other for certain datasets. In the dataset in Figure 2A, the stimuli is clustered such that the Euclidean distance between any two stimuli is correlated with their cosine distance. However, for the dataset in Figure 2B, the Euclidean distance is no longer correlated with the angle and the confounds lead to different representational geometries. Thus, this example illustrates how: (i) two systems acting on very different features of inputs can nevertheless end up with similar representational geometries when these features are able to mimic each other, and (ii) when the two systems are non-identical, the correlation in representational geometries will be modulated by the structure of the data – two systems may show a high correlation in their representational geometries on one set but a low correlation on another set.

Simulation 2: RSA between systems with different feature encodings

Simulation 1 made a number of simplifying assumptions – the dataset was two-dimensional, clustered into two categories and we intentionally chose functions Φ_1 and Φ_2 such that the kernel matrices were correlated in one case and not correlated in the other. It could be argued that, even though the above results hold in principle, they are unlikely in practice when the transformations and data structure are more complex. Indeed, it is possible that a similarity in representational geometries becomes less likely as one increases the number of categories (i.e., clusters or conditions) being considered.

To address this objection, we now consider a more complex setup, where the transformations Φ_1 and Φ_2 are modelled as feedforward deep neural networks (DNNs), trained to classify a high-dimensional dataset into multiple categories. Many studies that use RSA compare systems using naturalistic images as visual inputs [4, 10]. While using naturalistic images brings research closer to the real-world, it is also well-known that datasets of naturalistic images frequently contain confounds – independent features that can predict image categories [13]. We will now show how the simplest of such confounds, a single pixel, can lead to a high RSA between two DNNs that encode qualitatively different features of inputs.

Consider the same setup as above, where an input stimulus, $\boldsymbol{x}$, is transformed to a representation space by two systems, Φ_1 and Φ_2. Instead of a two-dimensional input space, $\boldsymbol{x}$ now exists in a high-dimensional image space and Φ_1 and Φ_2 are two versions of a DNN – `VGG-16` – trained to classify input images into different categories. We ensured that Φ_1 and Φ_2 were qualitatively different transformations of input stimuli by making the networks sensitive to different predictive features within the stimuli. The first network was trained

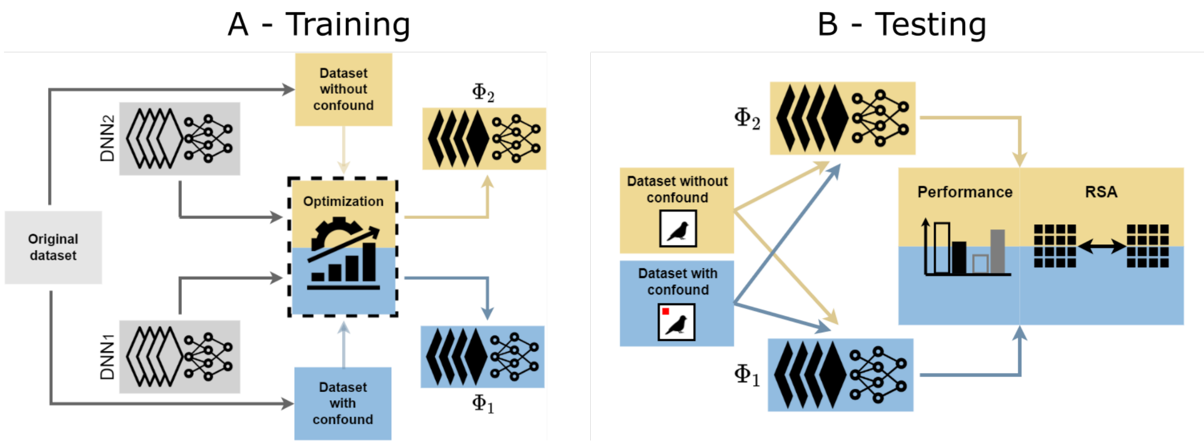

Figure 3: **Training and testing DNNs with different feature encodings.** Panel A shows the training procedure for Simulations 2–4, where we created two versions of the original dataset (gray), one containing a confound (blue) and the other left unperturbed (yellow). These two datasets were used to train two networks (gray) on a categorisation task, resulting in two networks that learn to categorise images either based on the confound (projection Φ_2) or based on statistical properties of the unperturbed image (projection Φ_1). Panel B shows the testing procedure where each network was tested on stimuli from each dataset – leading to a 2x2 design. Performance on these datasets was used to infer the features that each network encoded and their internal response patterns were used to calculate RSA-scores between the two networks.

on an unperturbed dataset, while the second network was trained on a modified version of the dataset, where each image was modified to contain a confound – a single pixel in a location that was diagnostic of the category (see Figure 3 for the general approach).

The locations of these diagnostic pixels were chosen such that they were correlated to the corresponding representational distances between classes in Φ_1. Our hypothesis was that if the representational distances in Φ_2 preserve the physical distances of diagnostic pixels in input space, then this confound will end up mimicking the representational

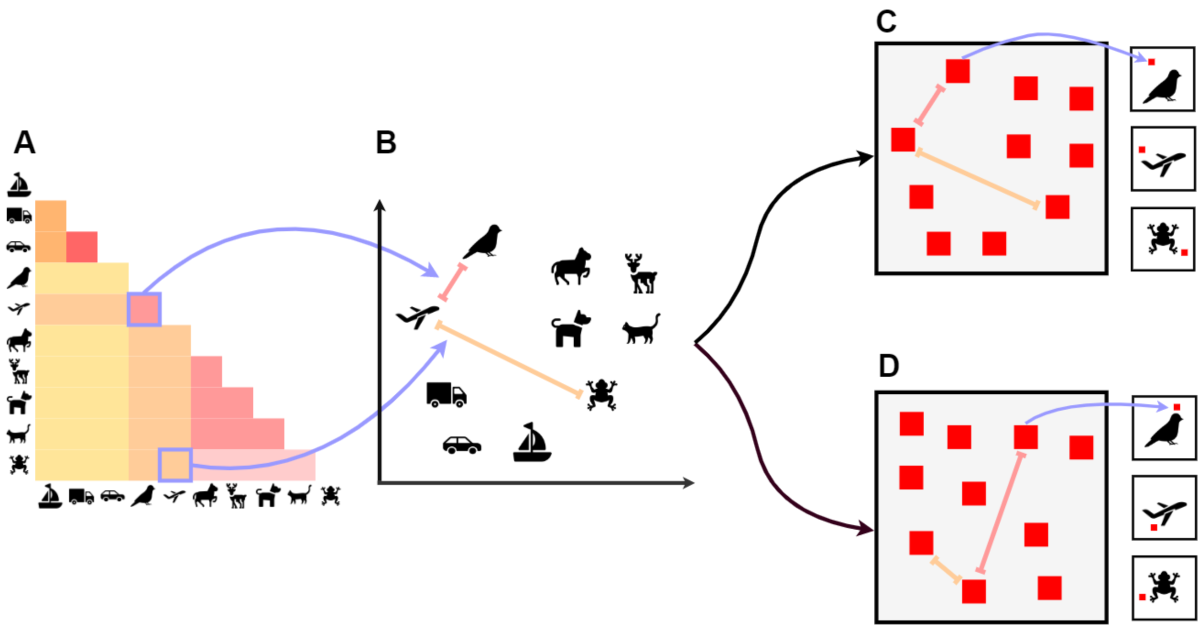

Figure 4: **Simulation 2 confound placement.** The representational geometry (Panel A and B) from the network trained on the unperturbed `CIFAR-10` images is used to determine the location of the single pixel confound (shown as a red patch here) for each category. In the 'Positive' condition (Panel C), we determined 10 locations in a 2D plane such that the distances between these locations were positively correlated to the representational geometry – illustrated here as the red patches in Panel C being in similar locations to category locations in Panel B. These 10 locations were then used to insert a single diagnostic – i.e., category-dependent – pixel in each image (Insets in Panel C). A similar procedure was also used to generate datasets where the confound was uncorrelated (Panel D) or negatively correlated (not shown here) with the representational geometry of the network.

geometry of Φ_1, even though the two systems use qualitatively different features for classification. Furthermore, we trained two more networks, Φ_3 and Φ_4, which were identical to Φ_2, except these networks were trained on datasets where the location of the confound was uncorrelated (Φ_3) or negatively correlated (Φ_4) with the representational distances in Φ_1 (see Figure 4 and Methods for details).

Classification accuracy (Figure 5 (left)) revealed that the network Φ_1, trained on the unperturbed images, learned to classify these images and ignored the diagnostic pixel – that is, it's performance was identical for the unperturbed and modified images. In contrast, networks Φ_2 (positive), Φ_3 (uncorrelated) and Φ_4(negative) failed to classify the unperturbed images (performance was statistically at chance) but learned to perfectly classify the modified images, showing that these networks develop qualitatively different representations compared to normally trained networks.

Next we computed pairwise RSA scores between the representations at the last convolution layer of Φ_1 and each of Φ_2, Φ_3 and Φ_4 (Figure 5 (right)). When presented unperturbed test images, the Φ_2, Φ_3 and Φ_4 networks all showed low RSA scores with the normally trained Φ_1 network. However, when networks were presented with test images that included the predictive pixels, RSA varied depending on the geometry of pixel locations in the input space. When the geometry of pixel locations was positively correlated to the normally trained network, RSA scores approached ceiling (i.e., comparable to RSA scores between two normally trained networks). Networks trained on uncorrelated and negatively correlated pixel placements scored much lower.

These results mirror Simulation 1: we observed that it is possible for two networks (Φ_1 and Φ_2) to show highly correlated representational geometries even though these networks learn to classify images based on very different features. One may argue that this could be because the two networks could have learned similar representations at

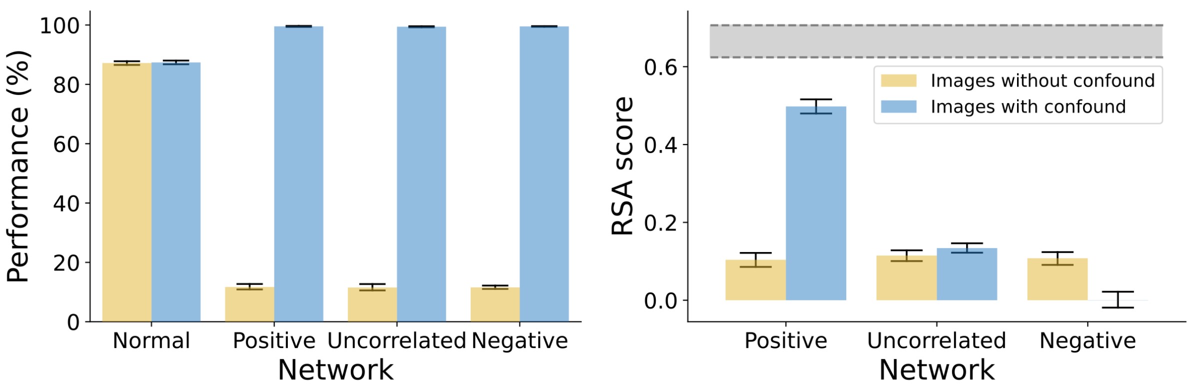

Figure 5: **Simulation 2 results.** *Left:* Performance of normally trained networks did not depend on whether classification was done on unperturbed `CIFAR-10` images or images with a single pixel confound (error bars represent 95% CI). All three networks trained on datasets with confounds could perfectly categorise the test images when they contained the confound (blue bars), but failed to achieve above-chance performance if the predictive pixel was not present (yellow bars). *Right:* The RSA score between the network trained on the unperturbed dataset and each of the networks trained on datasets with confounds. The three networks showed similar scores when tested on images without confounds, but vastly different RSA scores when tested on images with confounds. Networks in the Positive condition showed near ceiling scores (the shaded area represents noise ceiling) while networks in the Uncorrelated and Negative conditions showed much lower RSA.

the final convolution layer of the DNN and it is the classifier that sits on top of this representation that leads to the behavioural differences between these networks. But if this was true, it would not explain why RSA scores diminish for the two other comparisons (with Φ_3 and Φ_4). This modulation of RSA-scores for different datasets suggests that, like in Simulation 1, the correlation in representational geometry is not because the two systems encode similar features of inputs, but because different features mimic each other in their representational geometries.

Simulation 3: RSA between systems with different architectures

So far we have only considered high-dimensional systems with the same architecture – both Φ_1 and Φ_2 were DNNs that have the same set of units and learn through the same learning algorithm. Even though we observed that two systems that learn very different features can show a high correlation in their representational geometries, it could be argued that this is only possible because of the shared architecture and learning algorithm that underlies the two systems. On this veiw, a high RSA between systems that differ in architecture – e.g. a human and a macaque, or a DNN and a human – is unlikely unless both systems encode similar features of their inputs.

We address this argument in our next simulation, which compares representational geometries between activations in a DNN and macaque visual cortex. The experimental setup was similar to Simulation 2. We used the same set of images that were shown to macaques by [26] and modified this dataset to superimpose a small diagnostic patch on each image. In the same manner as in Simulation 2 above, we constructed three different datasets, where the locations of these diagnostic patches were either positively correlated, uncorrelated or negatively correlated with the RDM of macaque activations. We then trained four CNNs. The first CNN was pre-trained on `ImageNet` and then fine-

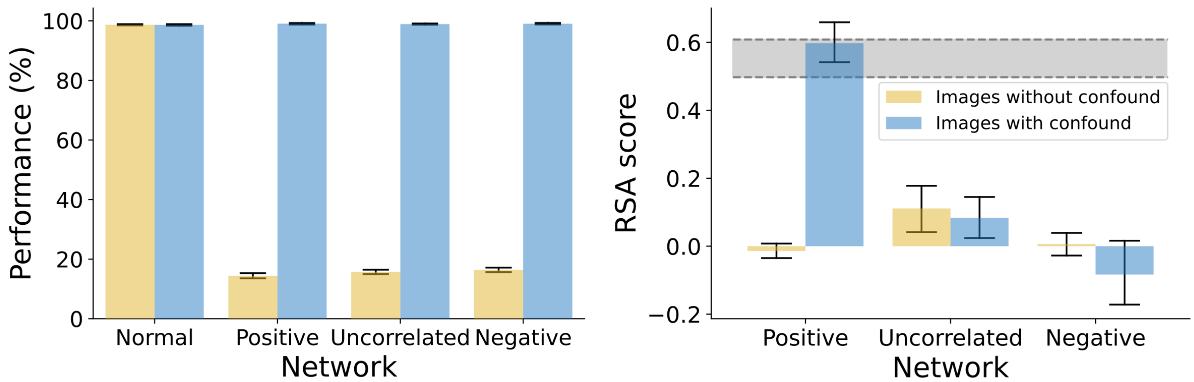

Figure 6: **Simulation 3 results.** *Left:* Classification Performance of the network trained on unperturbed images (Normal condition) did not depend on the presence or absence of the confound, while performance of networks trained with the confound (Positive, Uncorrelated and Negative conditions) highly depended on whether the confound was present. *Right:* RSA-scores with macaque IT activations were low for all three conditions when images did not contain a confound (yellow bars). When images contained a confound (blue bars), the RSA-scores depended on the condition, matching the RSA-score of the normally trained network (grey band) in the Positive condition, but decreasing significantly in the Uncorrelated and Negative conditions. The grey band represents a 95% CI for the RSA-score between normally trained networks and macaque IT activations.

tuned on the unmodified dataset of images shown to the macaques. Previous research has shown that CNNs trained in this manner develop representations that mirror the representational geometry of neurons in primate inferior temporal (IT) cortex [10]. The other three networks were trained on the three modified datasets and learned to entirely rely on the diagnostic patches (accuracy on images without the diagnostic patches was around chance).

Figure 6 (right) shows the correlation in representational geometry between the macaque

IT activations and activations at the final convolution layer for each of these networks. The correlation with networks trained on the unmodified images is our baseline and shown as the gray band in Figure 6. Our first observation was that a CNN trained to rely on the diagnostic patch can indeed achieve a high RSA score with macaque IT activations. In fact, the networks trained on patch locations that were positively correlated to the macaque RDM matched the RSA score of the CNNs trained on `ImageNet` and the unmodified dataset. This shows how two systems having very different architectures, encoding fundamentally different features of inputs (single patch vs naturalistic features) can show a high correspondence in their representational geometries. We also observed that, like in Simulations 2, the RSA score depended on the clustering of data in the input space – when patches were placed in other locations (uncorrelated or negatively correlated to macaque RDMs) the RSA score became significantly lower.

Simulation 4: RSA using structured datasets

All the simulations so far have used the same method to construct datasets with confounds – we established the representational geometry of one system (Φ_1) and constructed datasets where the clustering of features (pixels) mirrored this geometry. However, it could be argued that confounds which cluster in this manner are unlikely in practice. For example, even if texture and shape exist as confounds in a dataset, the inter-category distances between textures are not necessarily similar to the inter-category distances between shape.

However, categories in real-world datasets are usually hierarchically clustered into higher-level and lower-level categories. For example, in the `CIFAR-10` dataset, the Dogs and Cats (lower-level categories) are both animate (members of a common higher-level category) and Airplanes and Ships (lower-level categories) are both inanimate (members of a higher-level category). Due to this hierarchical structure, Dog and Cat images are

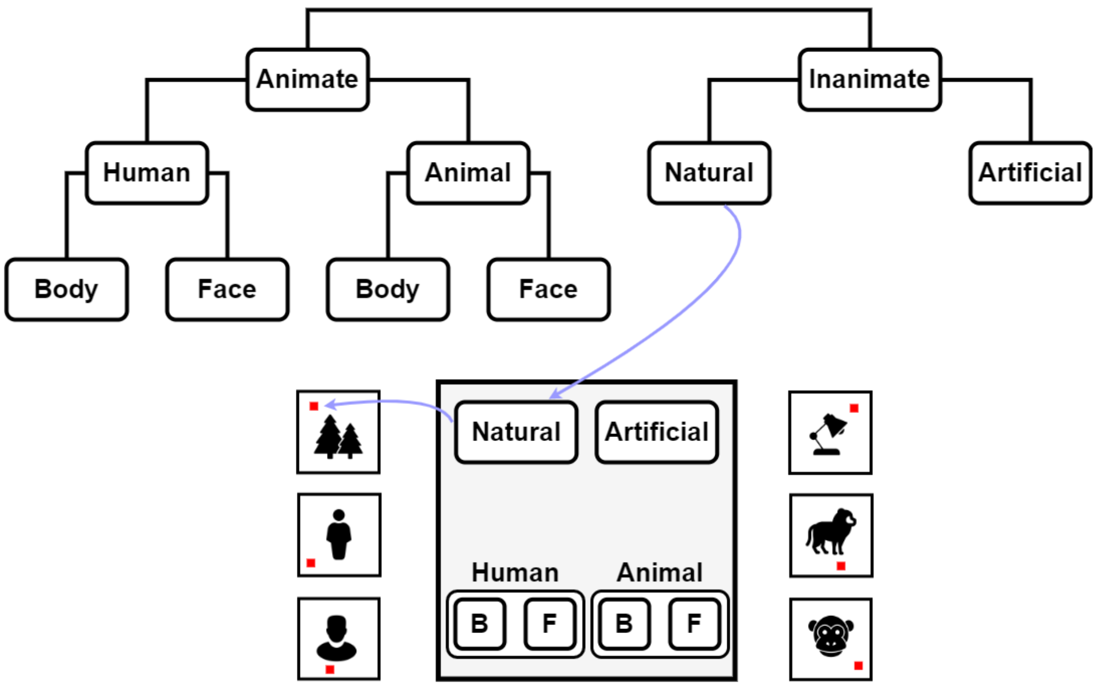

Figure 7: **Exploiting intrinsic dataset hierarchy in order to place confounds.** The top panel shows the hierarchical structure of categories in the dataset, which was used to place the single pixel confounds. The example at the bottom (middle) shows one such hierarchical placement scheme where the pixels for Inanimate images were closer to the top of the canvas while Animate images were closer to the bottom. Within the Animate images, the pixels for Humans and Animals were placed at the left and right, respectively, and the pixels for bodies (B) and faces (F) were clustered as shown.

likely to be closer to each other not only in their shape, but also their colour and texture (amongst other features) than they are to Airplane and Ship images. In our next simulation, we explore whether this hierarchical structure of categories can lead to a correlation in representational geometries between two systems that learn different feature encodings.

For this simulation, we selected a popular dataset used for comparing representational geometries in humans, macaques and deep learning models [11, 27]. This dataset consists of six categories which can be organised into a hierarchical structure shown in Figure 7. [4] showed a striking match in RDMs for response patterns elicited by these stimuli in human and macaque IT. For both humans and macaques, distances in response patterns were larger between the higher-level categories (animate and inanimate) than between the lower-level categories (e.g., between human bodies and human faces).

We used a similar experimental paradigm to the above simulations, where we trained networks to classify stimuli which included a single predictive pixel. But instead of using an RDM to compute the location of a diagnostic pixel, we used the hierarchical categorical structure. In the first modified version of the dataset, the location of the pixel was based on the hierarchical structure of categories in Figure 7 – predictive pixels for animate kinds were closer to each other than to inanimate kinds, and pixels for faces were closer to each other than to bodies, etc. One such configuration can be seen in Figure 7. In the second version, the predictive pixel was placed at a random location for each category (but, of course, at the same location for all images within each category). We call these conditions 'Hierarchical' and 'Random'. [11] showed that the RDM of average response patterns elicited in the human IT cortex (Φ_1) correlated with the RDM of a DNN trained on naturalistic images (Φ_2). We explored how this compared to the correlation with the RDM of a network trained on the Hierarchical pixel placement (Φ_3) and Random pixel placement (Φ_4).

Results for this simulation are shown in Figure 8. We observed that representational geometry of a network trained on Hierarchically placed pixels (Φ_3) was just as correlated to the representational geometry of human IT responses (Φ_1) as a network trained on naturalistic images (Φ_2). However, when the pixel locations for each category were randomly chosen, this correlation decreased significantly. These results suggest that any confound in the dataset (including texture, colour or low-level visual information) that has distances

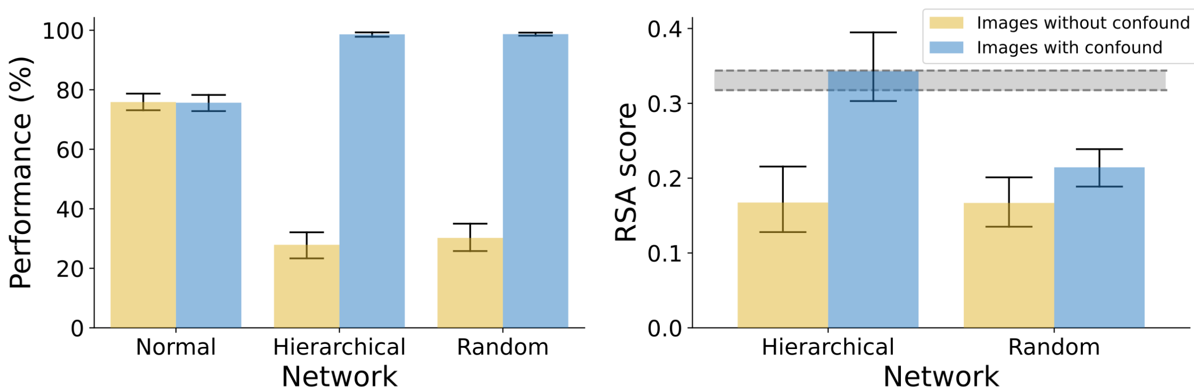

Figure 8: **Simulation 4 results.** *Left:* Performance of normally trained networks did not depend on whether the confound was present. Networks trained with the confound failed to classify stimuli without the confound (yellow bars) while achieving near perfect classification of stimuli with the confound present (blue bars). *Right:* RSA with human IT activations reveals that, when the confound was present, the RSA-score for networks in the Hierarchical condition matched the RSA-score of normally trained network (gray band), while the RSA-score of the network in the Random condition was significantly lower. The grey band represents 95% CI for the RSA score between normally trained networks and human IT.

governed by the hierarchical clustering structure of the data could underlie the observed similarity in representational geometries between CNNs and human IT. More generally, these results show how it is plausible that many confounds present in popular datasets may underlie the observed similarity in representational geometries between two systems. The error of inferring a similarity in mechanism based on a high RSA score is not just possible but also probable.

Discussion

In four simulations, we have illustrated a number of conditions under which it can be problematic to infer a similarity of representations between two systems based on a correlation in their representational geometries. We showed that two systems may transform their inputs through very different functions and encode very different features of inputs and yet have highly correlated representational geometries. In fact, we showed that this correlation can be a product of the structure of the dataset itself. A consequence of this result is that the RSA-score between two systems becomes dataset dependent. For example, one may observe a high RSA-score between a brain region of a primate and human for one dataset (e.g., [4]), but this score may become much lower for another dataset. Thus the observation of a similarity in representational geometry between systems must be interpreted with caution.

The significance of these results depends on whether you take an *externalist* or *holistic* view on mental representations. According to the first view, the content of representations is determined by their relationship to entities in the external world. This perspective is implicitly taken by most neuroscientists and psychologists, who are interested in comparing mechanisms underlying cognitive processes – that is, they are interested in the set of nested functions and algorithms responsible for transforming sensory input into a set of activations in the brain. From this perspective, our finding that high RSAs can be obtained between systems that work in qualitatively different ways poses a challenge to researchers using RSAs to compare systems.

Of course, a researcher with an externalist perspective may acknowledge that a second-order isomorphism of activity patterns does *not* strictly imply that two systems are similar mechanistically but still assume that it is highly likely to be the case. That is, as a practical matter, a researcher may assume that RSAs are a reliable method to compare systems. However, our findings challenge this assumption. We show how a high RSA between different systems can not only occur in principle, but also in practice, in high-dimensional systems operating on high-dimensional data. Indeed, we show that the hierarchical structure of datasets frequently used to test similarity of representations lends itself to a high RSA arising because of confounds present in the dataset. Such confounds are commonly found in high-dimensional stimuli such as naturalistic images that are frequently used to measure RSA [10, 27]. Indeed, presence of such confounds may explain why researchers have observed high RSAs between DNNs that classify objects based on texture [15, 16] and the human visual system that classifies by shape [21, 28].

Alternatively, a researcher may reject an externalist view and adopt the perspective that representations obtain their meaning based on how they are related to each other within each system, rather than based on their relationship to entities in the external world. That is, "representation *is* the representation of similarities" [29]. From this perspective, as long as the two systems share the same relational distances between internal activations, one can validly infer that the two systems have similar representations. That is, a second-order isomorphism implies a similarity of representations, by definition. This view has been called *holism* in the philosophy of mind [30, 31] and is related to a similar idea of *meaning holism* in language, which is the idea that the meaning of a linguistic expression is determined by its relation to other expressions within a language [32, 33]. For example, Firth [34] (p. 11) writes: "you shall know a word by the company it keeps". More recently, Griffiths and Steyvers [35], and Griffiths, Steyvers, and Tenenbaum [36] have adopted meaning holism accounts of semantic representations in neural networks. Our results are *not* problematic for a researcher adopting this holistic perspective. However, our results show that adopting this view misses the information about differences in mechanistic processes that a psychologist or neuroscientist is frequently interested in, for instance, whether the visual system processes shape or texture (or the location of diagnostic pixels) in order to identify objects. Fodor and Lepore long ago criticized this philosophical stance [31, 37], and interestingly, this philosophical debate played an important part in the development of RSA (see Supplementary Information, Section A). Unfortunately, this debate has largely been ignored by researchers who use RSA as a method to compare similarity of systems.

We would also like to make it clear that the results here are not a blanket criticism of the RSA approach as currently practiced. A representational dissimilarity matrix (RDM) contains important information about the similarity structures of representations. Any mechanistically correct model of an individual or a species must capture this similarity structure. As such, RSA provides a benchmark for *rejecting* possible models. However, the above simulations show that RSA may be a misleading benchmark for *selecting* models – two systems may show similar representational geometries and yet work on very different transformations and features of input stimuli (for an in depth discussion about inferring similarity of causal mechanisms from similar outcomes see [38]).

A related point has been made by Kriegeskorte and Diedrichson [39] and Kriegeskorte and Wei [40], who point out that two systems may have the same representational geometry, even if they have a different activity profile over neurons. In this sense, the geometry loses the information about how information was distributed over a set of neurons. Kriegeskorte and Diedrichson [39] equate this loss in information to "peeling a layer of an onion" – downstream decoders that are sensitive to the representational geometry rather than activity profiles over neuron populations can focus on difference in information as reflected by a change in geometry and be agnostic to how this information is distributed over a set of neurons. We agree that this invariance over activity profiles is indeed a useful property of representational geometries for downstream decoders. However, we are not aware of any studies that highlight how representational geometries also abstract over behaviourally relevant stimulus properties (e.g. shape vs texture). While abstracting over activity profiles may be useful, abstracting over stimulus properties loses an important piece of information when comparing representations across brain regions, individuals, species and between brains and computational models. Our simulations show how two systems may appear similar based on their representational geometries in one circumstance (e.g. Figure 2A) but drastically different in another circumstance (Figure 2B).

The key implication of our findings is that researchers should assess RSAs on a wider variety of datasets when comparing systems. Two systems that have the similar representations should show a high RSA irrespective of the stimuli on which they are tested, and testing systems on multiple datasets will reduce the likelihood that confounds or other factors are driving the effects. In practice, observing high RSAs after testing very different datasets, and datasets manipulated to avoid possible confounds, should be required before drawing strong conclusions regarding the similarity of two systems. In this regard, the "controversial stimuli" – images on which different computational models produce distinct responses – developed by [41] is a step in the right direction. By testing on stimuli that produce distinct responses in different models, one can adjudicate between models by comparing their representational geometries to the representational geometry of a target system. Combining RSA results with a range of methods, including experimental studies that stringently test hypotheses about how different systems work, seems the best approach going forward.

Methods

Dataset generation and training

All DNN simulations (Simulations 2–4) were carried out using the `Pytorch` framework [42]. The model implementations were downloaded from the `torchvision` library. Networks trained on unperturbed datasets in all simulations were pre-trained on `ImageNet` as were networks trained on modified datasets in Simulation 2. Networks trained on modified datasets in Simulations 3 and 4 were randomly initialised. For the pre-trained models, their pre-trained weights were downloaded from `torchvision.models` subpackage.

Simulation 1 Each dataset in Simulation 1 consists of 100 samples (50 in each cluster) drawn from two multivariate Gaussians, $\mathcal{N}(x|\mu, \mathbf{\Sigma})$, where μ is a 2-dimensional vector and $\mathbf{\Sigma}$ is a 2×2 covariance matrix. In Figure 2A, the two Gaussians have means $\mu_{\mathbf{1}} = (1, 8)$ and $\mu_{\mathbf{2}} = (8, 1)$ and a covariance matrices $\mathbf{\Sigma_1} = \mathbf{\Sigma_2} = \frac{1}{2}\mathbf{I}$, while in Figure 2B the Gaussians have means $\mu_{\mathbf{1}} = (1, 1)$ and $\mu_{\mathbf{2}} = (8, 8)$ and a covariance matrices $\mathbf{\Sigma_1} = \mathbf{I}$, $\mathbf{\Sigma_2} = 8\mathbf{I}$. All kernel matrices were computed using the `sklearn.metrics.pairwise` module of the `scikit-learn` Python package.

Simulation 2 First, a VGG-16 deep convolutional neural network [43], pre-trained on the `ImageNet` dataset of naturalistic images, was trained to classify stimuli from the `CIFAR-10` dataset [44]. The `CIFAR-10` dataset includes 10 categories with 5000 training, and 1000 test images per category. The network was fine-tuned on `CIFAR-10` by replacing the classifier so that the final fully-connected layer reflected the correct number of target classes in `CIFAR-10` (10 for `CIFAR-10` as opposed to 1000 for `ImageNet`). Images were rescaled to a size of 224×224px and then the model learnt to minimise the cross-entropy error using the RMSprop optimizer with a mini-batch size of 64, learning rate of 10^{-5}, and momentum of 0.9. All models were trained for 10 epochs, which were sufficient for convergence across all datasets.

Second, 100 random images from the test set for each category were sampled as input for the network and activations at the final convolutional layer extracted using the `THINGSVision` Python toolkit [45]. The same toolkit was used to generate a representational dissimilarity matrix (RDM) from the pattern of activations using `1-Pearson's r` as the distance metric. The RDM was then averaged by calculating the median distance between each instance of one category with each instance of the others (e.g., the median distance between `Airplane` and `Ship` was the median of all pair-wise distances between activity patterns for airplane and ship stimuli). This resulted in a 10×10, category-level, RDM which reflected average between-category distances.

Third, three modified versions of the `CIFAR-10` datasets were created for the 'Positive', 'Uncorrelated' and 'Negative' conditions, respectively. In each dataset, we added one diagnostic pixel to each image, where the location of the pixel depended on the category (See Figure 4). The locations of these pixels were determined using the averaged RDM from the previous step. We call this the target RDM. In the 'Positive' condition, we wanted the distances between pixel placements to be positively correlated to the distances between categories in the target RDM. We achieved this by using an iterative algorithm that sampled pixel placements at random, calculated an RDM based on distances between the pixel placements and computed an RSA-score (Spearman correlation) with the target RDM. Placements with a score above 0.70 were retained and further optimized (using small perturbations) to achieve an RSA-score over 0.90. The same procedure was also used to determine placements in the Uncorrelated (optimizing for a score close to 0) and Negatively correlated (optimizing for a negative score) conditions.

Finally, datasets were created using 10 different placements in each of the three conditions. Networks were trained for classification on these modified `CIFAR-10` datasets in the same manner as the VGG-16 network trained on the unperturbed version of the dataset (See Figure 3).

Simulation 3 The procedure mirrored Simulation 2 with the main difference being that the target system was the macaque inferior temporal cortex. Neural data from two macaques, as well as the dataset were obtained from the Brain Score repository [46]. This dataset consists of 3200 images from 8 categories (animals, boats, cars, chairs, faces, fruits, planes, and tables), we computed an 8×8 averaged RDM based on macaque IT response patterns for stimuli in each category.

This averaged RDM was then used as the target RDM in the optimization procedure to determine locations of the confound (here, a white predictive patch of size 5×5 pixels) for each category. Using a patch instead of a single pixel was required in this dataset because of the structure and smaller size of the dataset (3200 images, rather than 50,000 images for `CIFAR-10`). In this smaller dataset, the networks struggle to learn based on a single pixel. However, increasing the size of the patch makes these patches more predictive and the networks are able to again learn entirely based on this confound (see results in Figure 5). In a manner similar to Simulation 2, this optimisation procedure was used to construct three datasets, where the confound's placement was positively correlated, uncorrelated or negatively correlated with the category distances in the target RDM.

Finally, each dataset was split into 75% training (2432 images) and 25% test sets (768 images) before VGG-16 networks were trained on the unperturbed and modified datasets in the same manner as in Simulation 2. One difference between Simulations 2 and 3 was that here the networks in the Positive, Uncorrelated and Negative conditions were trained from scratch, i.e., not pre-trained on `ImageNet`. This was done because we wanted to make sure that the network in the Normal condition (trained on `ImageNet`) and the networks in the Positive, Uncorrelated and Negative conditions encoded fundamentally different features of their inputs – i.e., there were no `ImageNet`-related features encoded by representations Φ_2, Φ_3 and Φ_4 that were responsible for the similarity in representational geometries between these representations and the representations in macaque IT cortex.

Simulation 4 The target system in this simulation was human IT cortex. The human RDM and dataset were obtained from [4]. Rather than calculating pixel placements based on the human RDM, the hierarchical structure of the dataset was used to place the pixels manually. The dataset consists of 910 images from 6 categories: human bodies, human faces, animal bodies, animal faces, artificial inanimate objects and natural inanimate objects. These low-level categories can be organised into the hierarchical structure shown in Figure 7. Predictive pixels were manually placed so that the distance between pixels for Animate kinds were closer together than they were to Inanimate kinds and that faces were closer together than bodies. This can be done in many different ways, so we created five different datasets, with five possible arrangements of predictive pixels. Results in the Hieararchical condition (Figure 8) are averaged over these five datasets. Placements for the Random condition were done similarly, except that the locations were selected randomly.

Networks were then trained on a 6-way classification task (818 training images and 92 test images) in a similar manner to the previous simulations. As in Simulation 3, networks trained on the modified datasets (both Hierarchical and Random conditions) were not pre-trained on `ImageNet`.

RDM and RSA computation

For Simulations 2-4 all image-level RDMs were calculated using $1 - r$ as the distance measure. RSA scores were computed as the Spearman rank correlation between RDMs.

In Simulation 2, a curated set of test images was selected due to the extreme heterogeneity of the `CIFAR-10` dataset (low activation pattern similarity between instances of the same category). This was done by selecting 5 images per category which maximally correlated with the averaged activation pattern for the category. Since `CIFAR-10` consists of 10 categories, the RSA-scores in Simulation 2 were computed using RDMs of size 50×50.

In Simulation 3, the dataset consisted of 3200 images belonging to 8 categories. We first calculated a full 3200×3200 RDM using the entire set of stimuli. An averaged, category-level, 8×8 RDM was then calculated using median distances between categories (in a manner similar to that described for Simulation 2 in the Section 'Dataset generation and training'). This 8×8 RDM was used to determine the RSA-scores. We also obtained qualitatively similar results using the full 3200×3200 RDMs. These results can be found in the Supplementary Information, Section B.

In Simulation 4, the dataset consisted of 818 training images and 92 test images. Kriegeskorte et al. [4] used these images to obtain a 92×92 RDM to compare representations between human and macaque IT cortex. Here we computed a similar 92×92 RDM for networks trained in the Normal, Hierarchical and Random training conditions, which were then compared with the 92×92 RDM from human IT cortex to obtain RSA-scores for each condition.

Testing

In Simulation 2, we used a 4×2 design to measure classification performance for networks in all four conditions (Normal, Postive, Uncorrelated and Negative) on both unperturbed images and modified images. We computed six RSA-scores: three pairs of networks – Normal-Positive, Normal-Uncorrelated and Normal-Negative – and two types of inputs – unperturbed and modified test images. The noise ceiling (grey band in Figure 5) was determined in the standard way as described in [47] and represents the expected range of the highest possible RSA score with the target system (network trained on the unperturbed dataset).

In Simulation 3, performance was estimated in the same manner as in Simulation 2 (using a 4×2 design), but RSA-scores were computed between RDMs from macaque IT activations and the four types of networks – i.e. for the pairs Macaque-Normal, Macaque-Positive, Macaque-Uncorrelated and Macaque-Negative. And like in Simulation 2, we determined each of these RSA-scores for both unperturbed and modified test images as inputs to the networks.

In Simulation 4, performance and RSA were computed in the same manner as in Simulation 3, except that the target RDM for RSA computation came from activations in human IT cortex and the networks were trained in one of three conditions: Normal, Hierarchical and Random.

Data analysis

Performance and RSA scores were compared by running analyses of variance and Tukey HSD post-hoc tests. In Simulations 2 and 3, performance differences were tested by running a 4 (type of training) by 2 (type of dataset) mixed ANOVAs. In, Simulation 4, the differences were tested by running a 3x2 mixed ANOVA.

RSA scores with the target system between networks in various conditions were compared by running 3x2 ANOVAs in Simulations 2 and 3, and a 2x2 ANOVA in Simulation 4. We observed that RSA-scores were highly dependent on both the way the networks were trained and also the test images used to elicit response activations.
For a detailed overview of the statistical analyses and results, see Supplemental Information Section C.

Data Availability

Confound placement coordinates (all simulations), unperturbed datasets (Simulations 3 and 4), macaque activation patterns and RDMs (Simulation 3) and human RDM (Simulation 4) are available at OSF.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 741134)

References

[1] Hubel, D. H. & Wiesel, T. N. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology* **148**, 574–591 (1959).

[2] O'Keefe, J. Place units in the hippocampus of the freely moving rat. *Experimental Neurology* **51**, 78–109 (1976).

[3] Kriegeskorte, N., Mur, M. & Bandettini, P. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience* **2** (2008).

[4] Kriegeskorte, N. *et al.* Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron* **60**, 1126–1141 (2008).

[5] Haxby, J. V., Gobbini, M. I. & Nastase, S. A. Naturalistic stimuli reveal a dominant role for agentic action in visual representation. *NeuroImage* **216**, 116561 (2020).

[6] O'Hearn, K., Larsen, B., Fedor, J., Luna, B. & Lynn, A. Representational similarity analysis reveals atypical age-related changes in brain regions supporting face and car recognition in autism. *NeuroImage* **209**, 116322 (2020).

[7] Michael L. Mack, B. L., Alison R. Preston. Decoding the brain's algorithm for categorization from its neural implementation. *Current Biology* **23**, 2023–2027 (2013).

[8] Freund, M. C., Etzel, J. A. & Braver, T. S. Neural coding of cognitive control: The representational similarity analysis approach. *Trends in Cognitive Sciences* **25**, 622–638 (2021).

[9] Kaneshiro, B., Perreau Guimaraes, M., Kim, H.-S., Norcia, A. M. & Suppes, P. A representational similarity analysis of the dynamics of object processing using single-trial eeg classification. *PLOS ONE* **10**, 1–27 (2015).

[10] Yamins, D. L. K. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences* **111**, 8619–8624 (2014).

[11] Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLOS Computational Biology* **10**, 1–29 (2014).

[12] Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A. & Oliva, A. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports* **6**, 27755 (2016).

[13] Kietzmann, T. C. *et al.* Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences* **116**, 21854–21863 (2019).

[14] Kiat, J. E. *et al.* Linking patterns of infant eye movements to a neural network model of the ventral stream using representational similarity analysis. *Developmental Science* **25**, e13155 (2022).

[15] Geirhos, R. *et al.* Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* (2018).

[16] Geirhos, R. *et al.* Shortcut learning in deep neural networks. *Nature Machine Intelligence* **2**, 665–673 (2020).

[17] Hermann, K., Chen, T. & Kornblith, S. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems* **33** (2020).

[18] Malhotra, G., Evans, B. D. & Bowers, J. S. Hiding a plane with a pixel: examining shape-bias in cnns and the benefit of building in biological constraints. *Vision Research* **174**, 57–68 (2020).

[19] Malhotra, G., Dujmovic, M. & Bowers, J. S. Feature blindness: a challenge for understanding and modelling visual object recognition (in press). *PLOS Computational Biology, preprint bioRxiv:2021.10.20.465074* (2022).

[20] Navon, D. Forest before trees: The precedence of global features in visual perception. *Cognitive psychology* **9**, 353–383 (1977).

[21] Biederman, I. & Ju, G. Surface versus edge-based determinants of visual recognition. *Cognitive psychology* **20**, 38–64 (1988).

[22] Landau, B., Smith, L. B. & Jones, S. S. The importance of shape in early lexical learning. *Cognitive development* **3**, 299–321 (1988).

[23] Xu, Y. & Vaziri-Pashkam, M. Limits to visual representational correspondence between convolutional neural networks and the human brain. *Nature Communications* **12**, 2065 (2021).

[24] Schölkopf, B. & Smola, F., A. J.and Bach. *Learning with kernels: support vector machines, regularization, optimization, and beyond* (MIT Press, 2002).

[25] Sahami, M. & Heilman, T. D. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, 377–386 (Association for Computing Machinery, New York, NY, USA, 2006).

[26] Majaj, N. J., Hong, H., Solomon, E. A. & DiCarlo, J. J. Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. *Journal of Neuroscience* **35**, 13402–13418 (2015).

[27] Kriegeskorte, N. Relating population-code representations between man, monkey, and computational models. *Frontiers in Neuroscience* **3**, 363–373 (2009).

[28] Smith, L. B., Jones, S. S., Landau, B., Gershkoff-Stowe, L. & Samuelson, L. Object name learning provides on-the-job training for attention. *Psychological science* **13**, 13–19 (2002).

[29] Edelman, S. Representation is representation of similarities. *Behavioral and Brain Sciences* **21**, 449–467 (1998).

[30] Block, N. Advertisement for a semantics for psychology. *Midwest Studies in Philosophy* **10**, 615–678 (1986).

[31] Fodor, J. & Lepore, E. *Holism: A Shoppers Guide* (Blackwell, Cambridge, 1992).

[32] Hempel, C. G. Problems and changes in the empiricist criterion of meaning. *Revue Internationale de Philosophie* **4**, 41–63 (1950).

[33] Quine, W. V. Main trends in recent philosophy: Two dogmas of empiricism. *The Philosophical Review* **60**, 20–43 (1951).

[34] Firth, J. R. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis* (1957).

[35] Griffiths, T. L. & Steyvers, M. A probabilistic approach to semantic representation. In *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society* (Erlbaum, Hillsdale, NJ, 2002).

[36] Griffiths, T. L., Steyvers, M. & Tenenbaum, J. A probabilistic approach to semantic representation. *Psychological Review* **114**, 211–244 (2007).

[37] Fodor, J. *Psychosemantics: The Problem of Meaning in the Philosophy of Mind* (MIT Press, Cambridge, 1987).

[38] Guest, O. & Martin, A. E. On logical inference over brains, behaviour, and artificial neural networks. *PsyArXiv preprint: 10.31234/osf.io/tbmcg* (2021).

[39] Kriegeskorte, N. & Diedrichsen, J. Peeling the onion of brain representations. *Annual Review of Neuroscience* **42**, 407–432 (2019).

[40] Kriegeskorte, N. & Wei, X.-X. Neural tuning and representational geometry. *Nature Reviews Neuroscience* **22**, 703–718 (2021).

[41] Golan, T., Raju, P. C. & Kriegeskorte, N. Controversial stimuli: Pitting neural networks against each other as models of human cognition. *Proceedings of the National Academy of Sciences* **117**, 29330–29337 (2020).

[42] Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, 8024–8035 (Curran Associates, Inc., 2019).

[43] Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[44] Krizhevsky, A. Learning multiple layers of features from tiny images. Tech. Rep. (2009).

[45] Muttenthaler, L. & Hebart, M. N. Thingsvision: A python toolbox for streamlining the extraction of activations from deep neural networks. *Frontiers in Neuroinformatics* **15**, 679838 (2021).

[46] Schrimpf, M. *et al.* Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv preprint: 407007* (2018).

[47] Nili, H. *et al.* A toolbox for representational similarity analysis. *PLOS Computational Biology* **10**, 1–11 (2014).