



Non-uniqueness Phenomenon of Object Representation in Modeling IT Cortex by Deep Convolutional Neural Network (DCNN)

Qiulei Dong^{1,2,3}, Bo Liu^{1,2} and Zhanyi Hu^{1,2,3*}

¹ National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, ² School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, ³ Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China

Recently DCNN (Deep Convolutional Neural Network) has been advocated as a general and promising modeling approach for neural object representation in primate inferotemporal cortex. In this work, we show that some inherent non-uniqueness problem exists in the DCNN-based modeling of image object representations. This non-uniqueness phenomenon reveals to some extent the theoretical limitation of this general modeling approach, and invites due attention to be taken in practice.

Keywords: deep convolutional neural network, neural object representation, inferotemporal cortex, non-uniqueness, image object representation

OPEN ACCESS

Edited by:

Kenway Louie,
New York University, United States

Reviewed by:

Guy Elston,
University of the Sunshine Coast,
Australia

Qiang Luo,
Fudan University, China

*Correspondence:

Zhanyi Hu
huzy@nlpr.ia.ac.cn

Received: 20 November 2019

Accepted: 09 April 2020

Published: 12 May 2020

Citation:

Dong Q, Liu B and Hu Z (2020)
Non-uniqueness Phenomenon of
Object Representation in Modeling IT
Cortex by Deep Convolutional Neural
Network (DCNN).
Front. Comput. Neurosci. 14:35.
doi: 10.3389/fncom.2020.00035

1. INTRODUCTION

Object recognition is a fundamental task of a biological vision system. It is widely believed that the primate inferotemporal (IT) cortex is the final neural site for visual object representation. Due to viewpoint change, illumination variation and other factors, how visual objects are represented in IT cortex, which manifests sufficient invariance to such identity-orthogonal factors, is still largely an open issue in neuroscience.

There are many different natural and manmade object categories, and each category in turn contains various different members. Currently, a number of works in neuroscience advocate the DCNN (Deep Convolutional Neural Network) as a new framework for modeling vision and brain information processing (Cadieu et al., 2014; Khaligh and Kriegeskorte, 2014; Kriegeskorte, 2015). In Yamins et al. (2014), Yamins and DiCarlo (2016), DCNN is regarded as a promising general modeling approach for understanding sensory cortex, called “the goal-driven approach.”

The basic idea of the goal-driven approach for IT cortex modeling can be summarized as: *a multi-layered DCNN is trained by ONLY optimizing the object categorization performance with a large set of visual category-labeled objects. Once a high categorization performance is achieved, the outputs of the penultimate layer neurons of the trained DCNN, which are regarded as the object representation, can reliably predict the IT neuron spikes for other visual stimuli in rapid object recognition*¹. In addition, the outputs of the upstream layer neurons can also predict the V4 neuron spikes. The goal-driven approach is conceptually eloquent and has been successfully used to model IT cortex in rapid object recognition and predict category-orthogonal properties (Hong et al., 2016).

¹The goal-driven approach is for modeling IT neuron representation in rapid object vision, which is assumed largely a feed forward process, hence could be modeled by DCNNs which are also feed forward networks.

2. DOES THE GOAL-DRIVEN APPROACH SATISFY THE UNIQUENESS REQUIREMENT IN MODELING IT CORTEX?

2.1. Motivation

Although some experimental results have demonstrated the success of the goal-driven approach in modeling IT cortex to some extent as mentioned above, the following uniqueness problem on the fundamental premise of the goal-driven approach is still unclear: does there exist a unique pattern of activations of the neurons (units) in the penultimate layer of a DCNN to a given set of image stimuli by only optimizing the object categorization performance? This uniqueness problem on object representation via a DCNN has a great influence on the theoretical foundation and generality of the goal-driven approach in particular, and the DCNN as a new framework for vision modeling in general.

In this work, we aim to provide a theoretical analysis on this problem as well as some supporting experimental results. Note that our current work is to clarify the non-uniqueness problem in object representation modeling with DCNNs under the goal-driven approach, it does not mean DCNNs could account for IT diverse specifications, as revealed in numerous works (Elston, 2002, 2007; Jacobs and Scheibel, 2002; Spruston, 2008; Elston and Fujita, 2014; Luebke, 2017).

In order to analyse this problem more clearly, we firstly introduce the definition of DCNN layer's object representation as used for predicting the neuron responses of primate IT cortex in the aforementioned goal-driven approach:

Definition 1. For a layer of a DCNN for object recognition, the activations of the neurons in this layer to an input object image is defined as its object representation.

Following the convention in the computational neuroscience, the following representation equivalence is introduced to evaluate whether the object representations learnt from two DCNNs are the same or not:

Definition 2. Given a set of object image stimuli, if the two object representations of two DCNNs on these stimuli can be related by a linear transformation, they are considered equivalent, or the same representations. Otherwise, they are different representations.

In the deep learning community, a recent active research topic is called "convergent learning" (Li et al., 2016), referring whether different DCNNs can learn the same representation at the level of neurons or groups of neurons. A generally reached conclusion is that different DCNNs with the same network architecture but trained only with different random initializations, have largely different representations at the level of neurons or groups of neurons, although their image categorization performances are similar. Note that although Li et al.'s work and the goal-driven approach focus on the representation from different points of view, the representations in the two works are closely related. Hence, the results in Li et al. (2016) could also re-highlight the aforementioned uniqueness problem in object representation via a DCNN to some extent.

Addressing this uniqueness problem, we show in the following section that, in theory, by only optimizing the image categorization accuracy, different DCNNs can give different object representations though they have exactly the same categorization accuracy. In other words, the obtained object representations by DCNNs under the goal-driven approach could be inherently non-unique, at least in theory.

2.2. Theoretical Analysis and Experimental Results

Proposition 1. *If the "Softmax" function is used as the final classifier for image categorization in modeling N categories of objects via a DCNN, and the object category with the largest probability is chosen as the final categorization, and if $x = (x_1, x_2, \dots, x_N)^T \in \mathbb{R}^N$ is the final output of this DCNN for an input image object I , $f(\cdot)$ is a univariate non-linear monotonically increasing function, $y \triangleq (y_1, y_2, \dots, y_N)^T = F(x) = (f(x_1), f(x_2), \dots, f(x_N))^T$, then x and y give exactly the same categorization result.*

Proof: For x and y , their corresponding probability vectors by Softmax are respectively:

$$C_x = \left(\frac{e^{x_1}}{\sum_{i=1}^N e^{x_i}}, \frac{e^{x_2}}{\sum_{i=1}^N e^{x_i}}, \dots, \frac{e^{x_N}}{\sum_{i=1}^N e^{x_i}} \right)^T \quad (1)$$

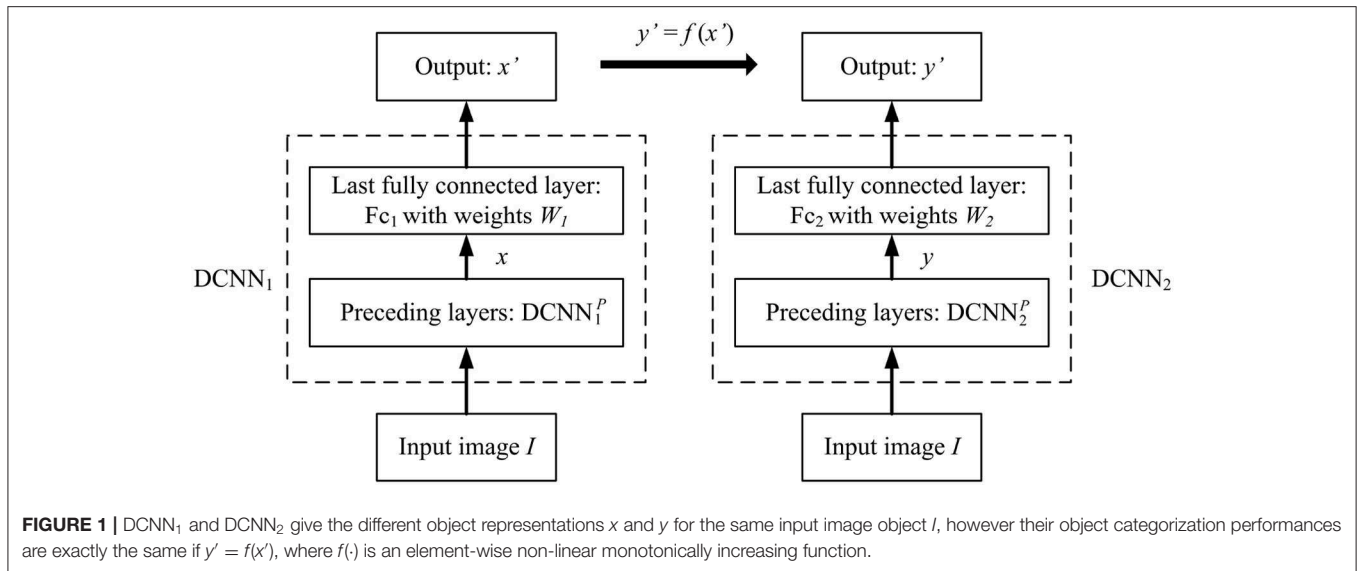
$$C_y = \left(\frac{e^{y_1}}{\sum_{i=1}^N e^{y_i}}, \frac{e^{y_2}}{\sum_{i=1}^N e^{y_i}}, \dots, \frac{e^{y_N}}{\sum_{i=1}^N e^{y_i}} \right)^T \quad (2)$$

Since $y_i = f(x_i)$ ($i = 1, 2, \dots, N$) and $f(\cdot)$ is a monotonically increasing function, the magnitude order of elements for x and y does not change. Then the magnitude order of the two probability vectors C_x and C_y does not change. Since the object category with the largest probability is chosen as the final categorization, both the indices of the largest elements in C_x and C_y are the same, hence the same categorization results are obtained for x and y . ■

Remark 1: Since $f(\cdot)$ is a non-linear function, x and y cannot be related by a linear transformation. In addition, in the deep learning community, the Softmax function is commonly used to convert the output vector of the network into a probability vector, and the category with the largest probability value is chosen as the final category.

Remark 2: In theory, $f(\cdot)$ could be different for different input image I . More generally, even the demand of monotonicity for $f(\cdot)$ is unnecessary, we need only the index of the largest value in y is the same to that in x because only the largest value determines the correct categorization. For the Top- K categorization accuracy, we need the index set of the K largest values in y keep the same to that in x , and the rest elements are not required. Hereinafter, for the notational convenience in discussion and practicality of implementation, we always assume $f(\cdot)$ is a univariate non-linear monotonically increasing function.

Proposition 2. *As shown in Figure 1, assume that $DCNN_1$ is a multi-layered network, concatenating a sub-network $DCNN_1^P$ whose output is x , and a fully connected layer with weight matrix*



$W_1 \in \mathbb{R}^{N \times M}$ and bias $b_1 \in \mathbb{R}^{N \times 1}$ ($\{M, N\}$ are the numbers of neurons at the penultimate layer and last layer of DCNN₁, respectively, with $M > N$), with $x' = W_1 x + b_1$. And assume that DCNN₂ is a multi-layered network, concatenating a sub-network DCNN₂^p whose output is y , and a fully connected layer with weight matrix $W_2 \in \mathbb{R}^{N \times M}$ and bias $b_2 \in \mathbb{R}^{N \times 1}$, with $y' = W_2 y + b_2$. If $y' = f(x')$ in element-wise mapping where $f(\cdot)$ is a monotonically increasing function, then the object representation x under DCNN₁ cannot be related by a linear transformation to the object representation y under DCNN₂, or x and y are two different object representations under the goal-driven approach.

Proof: Since $y' = f(x')$ in element-wise mapping where $f(\cdot)$ is a monotonically increasing function, according to Proposition 1, DCNN₁ and DCNN₂ have the identical image object categorization performance.

Since $x' = W_1 x + b_1$, then $x = (W_1^T W_1)^+ W_1^T (x' - b_1)$, where A^+ denotes the pseudo-inverse of matrix A . Similarly, $y = (W_2^T W_2)^+ W_2^T (y' - b_2)$. By Proposition 1, x' and y' is related by a non-linear function, then x and y cannot be related by a linear transformation either. In other words, x and y are two different object representations under the goal-driven approach. ■

Remark 3: Since $\{W_1, W_2\} \in \mathbb{R}^{N \times M}$ and $M > N$ in Proposition 2, the pseudo-inverse operator is used in the above proof. Here are a few words on the pseudo-inverse: Since $M > N$, which is the usual case in most existing DCNNs for object categorization (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015), the inverse $(W_i^T W_i)^+ (i = 1, 2)$ is not unique, but the equalities in $x = (W_1^T W_1)^+ W_1^T (x' - b_1)$ and $y = (W_2^T W_2)^+ W_2^T (y' - b_2)$ can be strictly met.

Proposition 2 indicates that given DCNN₁ with output x' , if there exists another multi-layered network DCNN₂ to output $y' = f(x')$, their representations x and y would be different but with identical categorization performance. This means that the aforementioned non-uniqueness problem in

object representation modeling under the goal-driven approach would arise regardless of how many training images are used, and how many exemplar images in each category are included. In other words, the non-uniqueness problem is an inherent problem in DCNN modeling under the goal-driven approach, and it cannot be completely removed by using more training data, at least in theory.

In the above, an implicitly assumption is that given a DCNN₁ with the output x'_i , there always exists a DCNN₂ with the output $y'_i = f(x'_i)$. Does such a DCNN₂ really always exist? This issue can be separately addressed for the following two cases. The first one is that DCNN₁ and DCNN₂ could be of different architectures, and the second one is that they are of the same architecture, but merely initialized differently during training.

2.2.1. The Different Architecture Case

Proposition 3. There always exists a multi-layered network to map I_i to y_i for the given input-output pairs $\{(I_i \leftrightarrow y_i), i = 1, 2, \dots, n\}$ in Proposition 2.

Proof: As shown in Proposition 2 and Figure 1, since DCNN₁ exists, it maps I to x . Denote this mapping function as $x = S_1(I) = DCNN_1^p(I)$. Since $x' = W_1 x + b_1$, $y' = F(x') = ((f(x'_1), f(x'_2), \dots, f(x'_n)))$, $y' = W_2 y + b_2$, and $y = (W_2^T W_2)^+ W_2^T (y' - b_2)$, we have:

$$\begin{aligned} y &= (W_2^T W_2)^+ W_2^T (y' - b_2) \\ &= (W_2^T W_2)^+ W_2^T (F(W_1 S_1(I) + b_1) - b_2) \end{aligned} \quad (3)$$

This is just the required mapping function. According to the Universal Approximation Theorem in Csáji (2001), it could be straightforwardly inferred that there always exists a DCNN with an arbitrary number $k + 1 (k \geq 1)$ of hidden layers, denoted as DCNN₂, whose sub-network DCNN₂^p with k hidden layers is able to approximate this function. ■

Proposition 3 indicates that given a DCNN₁, there always exists a DCNN₂ whose architecture may be different from

DCNN₁, so that the object representations of the two DCNNs are different but with the same categorization performance. A training procedure is described in the **Appendix**, to show how to train such a pair of DCNN₁ and DCNN₂.

Remark 4: In the proof, the only requirement for DCNN₂ is that it should have sufficient capacity to represent the input object set, but it does not necessarily have a similar network architecture to DCNN₁. Note that the sufficient representational capacity is an implicit necessary requirement for any DCNN-based applications.

Remark 5: In the proof, the number of input images is assumed to be unknown. However, for the finite-input case, Theorem 1 in Tian (2017) guarantees that there exists a two-layered neural network with ReLU activation and $(2n + d)$ weights, which could represent any mapping function from input to output on sample of size n in d dimensions. Of course, such a constructed network could be of a memorized neural network, i.e., it can ensure the given finite inputs to be mapped to the required outputs, but it cannot guarantee that the constructed network could possess sufficient generalization ability for new samples.

2.2.2. The Same Architecture Case

When DCNN₁ and DCNN₂ are obtained with the same network architecture but only trained under different random initializations, clearly a theoretical proof is impossible. However, based on the reported results in the “convergent learning” literatures as well as our simulated experimental results, it seems they still largely have non-equivalent object representations although they have similar categorization performances.

(1) Non-uniqueness results from “convergent learning” literatures

Using AlexNet (Krizhevsky et al., 2012) as a benchmark, Li et al. (2016) showed that by keeping the architecture unchanged but only trained with different random initializations, the obtained 4 DCNNs have similar categorization performances, but their object representations are largely different in terms of one-to-one, one-to-many, and many-to-many linear representation mapping. Note that the many-to-many mapping in Li et al. (2016) is closely related to the equivalence representation in Definition 2. Hence, the four representations are largely non-equivalent and this non-equivalence becomes more prevalent with increasing convolutional layers.

By introducing the concepts of “ ϵ -simple match set” and “ ϵ -maximum match set,” Wang et al. (2018) showed that for the 2 representative DCNNs, VGG (Simonyan and Zisserman, 2014), and ResNet (He et al., 2016), the size of maximum match set between the activation vectors of individual neurons at the same layer of the two DCNNs, which are also obtained with only different initializations as did in Li et al. (2016), is tiny compared with the number of the neurons at that layer. It was further found that only the outputs of neurons in the ϵ -maximum match set can be approximated within ϵ -error bound by a linear transformation, which indicates that for majority of the neurons at the same layer, their outputs cannot be reasonably approximated by a linear transformation, or the corresponding object representations are largely not equivalent.

(2) Non-uniqueness results from our experiments

Definition 3. If two DCNNs, DCNN₁ and DCNN₂, have similar image categorization performances with the same network architecture but different parameter configurations, they are called the similar performing pair of DCNNs.

Generally speaking, our results further confirm the non-uniqueness phenomenon of object representation under the goal-driven approach. We systematically investigated the representation differences between a similar performing pair of DCNNs on the two public object image datasets, CIFAR-10 that contains 60,000 images belonging to 10 categories of objects and CIFAR-100 that contains 60,000 images belonging to 100 categories of objects (Krizhevsky, 2009). In our experiments, 5,000 images per category in CIFAR-10 (also 500 images per category in CIFAR-100) were randomly selected for network training, and the rest for testing. Six network architectures with different configurations (denoted as {D1, D2, D3, D4, D5, D6}) were employed for evaluations, where {D1, D2, D3, D5, D6} were for CIFAR-10 and {D3, D4, D6} were for CIFAR-100 as shown in **Table 1**.

The traditionally used measure, “explained variance” (EV), was employed to access the degree of linearity between the learnt object representations from a similar performing pair of DCNNs, and we trained similar performing pairs of DCNNs under the following two schemes:

- Scheme-1: Both DCNN₁ and DCNN₂ were trained with random initializations.
- Scheme-2: Similar to the training procedure in the DCNN₁ was firstly trained with the Softmax loss, and then DCNN₂ was trained by combining the Softmax loss on the neuron outputs of the last layer and the Euclidean loss on the differences between the neuron outputs of the penultimate layer in DCNN₂ and the corresponding terms calculated according to Equation (3) (In our experiments, $f(x) = |x|\sqrt{x}$).

Here are some main results from our experiments:

(i) Explained variance on standard data

The results using the training Scheme-1 are shown in **Figure 2**. **Figures 2A,C** show the categorization accuracies of similar performing pairs of DCNNs under different network architectures with two random initializations on CIFAR-10 and CIFAR-100, respectively. The blue bars of **Figures 2B,D** show the corresponding mean EVs on CIFAR-10 and CIFAR-100, respectively. As seen from **Figures 2B,D**, the mean EVs by {D1, D2, D3, D5, D6} are around 63.4–87.5% on CIFAR-10, while the mean EVs by {D3, D4, D6} are around 53.6–65.9% on CIFAR-100. In addition, the mean EV of the network D1 under the training Scheme-2 is 51.2% on CIFAR-10.

Two points are revealed from these results:

- Given a similar performing pair of DCNNs, although the representations of the two DCNNs cannot in theory be related by a linear transformation, the explained variance between the two representations is relatively large.
- A similar performing pair of DCNNs with a deeper architecture, or having more layers, will generally have a

TABLE 1 | Network configurations (shown in columns).

ConvNet configuration					
D1 5 Layers	D2 8 Layers	D3 8 Layers	D4 8 Layers	D5 15 Layers	D6 9 Layers
Input (32*32 RGB Image)					
Conv5-32	Conv3-bn-32 Conv3-bn-32	Conv3-bn-64 Conv3-bn-64	Conv3-bn-128 Conv3-bn-128	Conv3-bn-32 Conv3-bn-32 Conv3-bn-32 Conv3-bn-32	Conv3-bn-64
Conv5-32	Conv3-bn-64 Conv3-bn-64	Conv3-bn-128 Conv3-bn-128	Conv3-bn-256 Conv3-bn-256	Conv3-bn-64 Conv3-bn-64 Conv3-bn-64 Conv3-bn-64	Conv3-bn-128
Conv5-64	Conv3-bn-128 Conv3-bn-128	Conv3-bn-256 Conv3-bn-256	Conv3-bn-512 Conv3-bn-512	Conv3-bn-128 Conv3-bn-128 Conv3-bn-128 Conv3-bn-128	Conv3-bn-256 Conv3-bn-256
Fc-64	Conv3-bn-256	Conv3-bn-512	Conv3-bn-1024	Conv3-bn-256 Conv3-bn-256	Conv3-bn-512 Conv3-bn-512
Fc-10	Fc-10	Fc-10(100)	Fc-100	Fc-10	Fc-10(100)

The convolutional layer parameters are denoted as "Conv(receptive field size)-bn-(number of channels)." The Fully connected layer parameters are denoted as "Fc-(number of units)".

larger explained variance between the two representations. The underlying reason seems that since a DCNN with a deeper architecture will generally have a larger representational capacity and since a fixed task has a fixed representation demand, a DCNN with a larger capacity will give a more linear representation.

In addition, for a similar performing pair, although their categorization performances are similar, it does not mean that the two DCNNs have the identical categorization label for each input sample, either correct or wrong. We have manually checked the categorization results for CIFAR-10 and CIFAR-100. The orange bars of **Figures 2B,D** show the computed mean EVs for only those inputs correctly categorized. As seen from **Figure 2**, the discrepancy of the explained variances between the representations of only the correctly categorized inputs and those of the whole inputs is insignificant and negligible in most cases, and it is perhaps due to the already high categorization rate of the two DCNNs such that the incorrectly categorized inputs only take a small fraction of a relatively large test set.

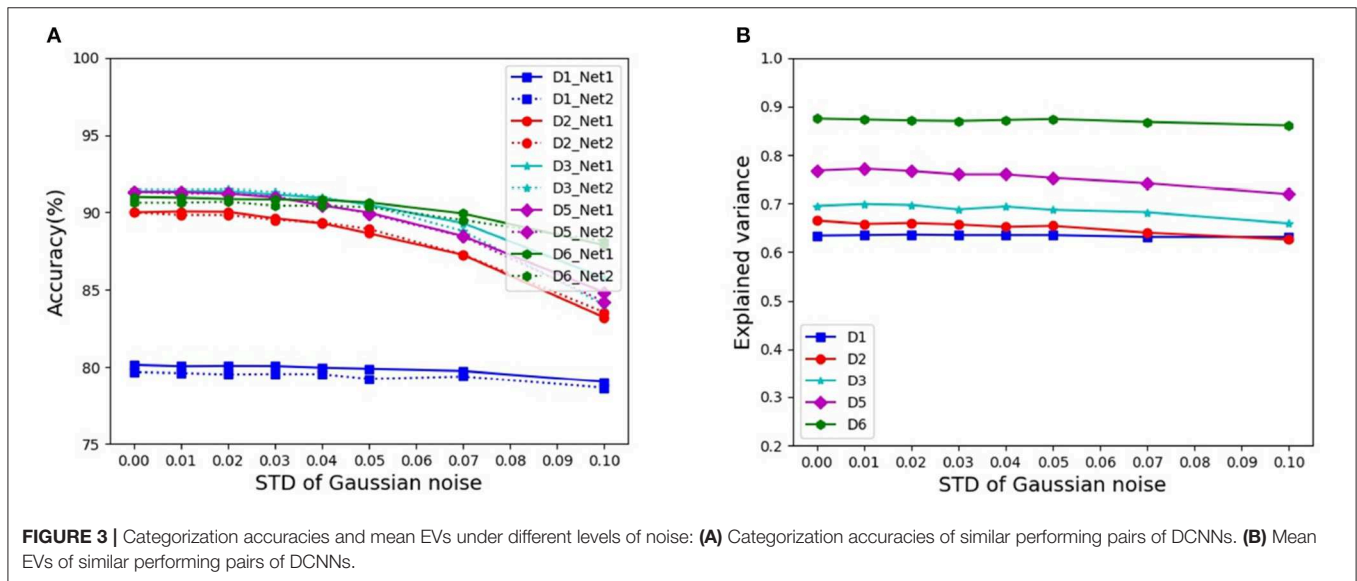
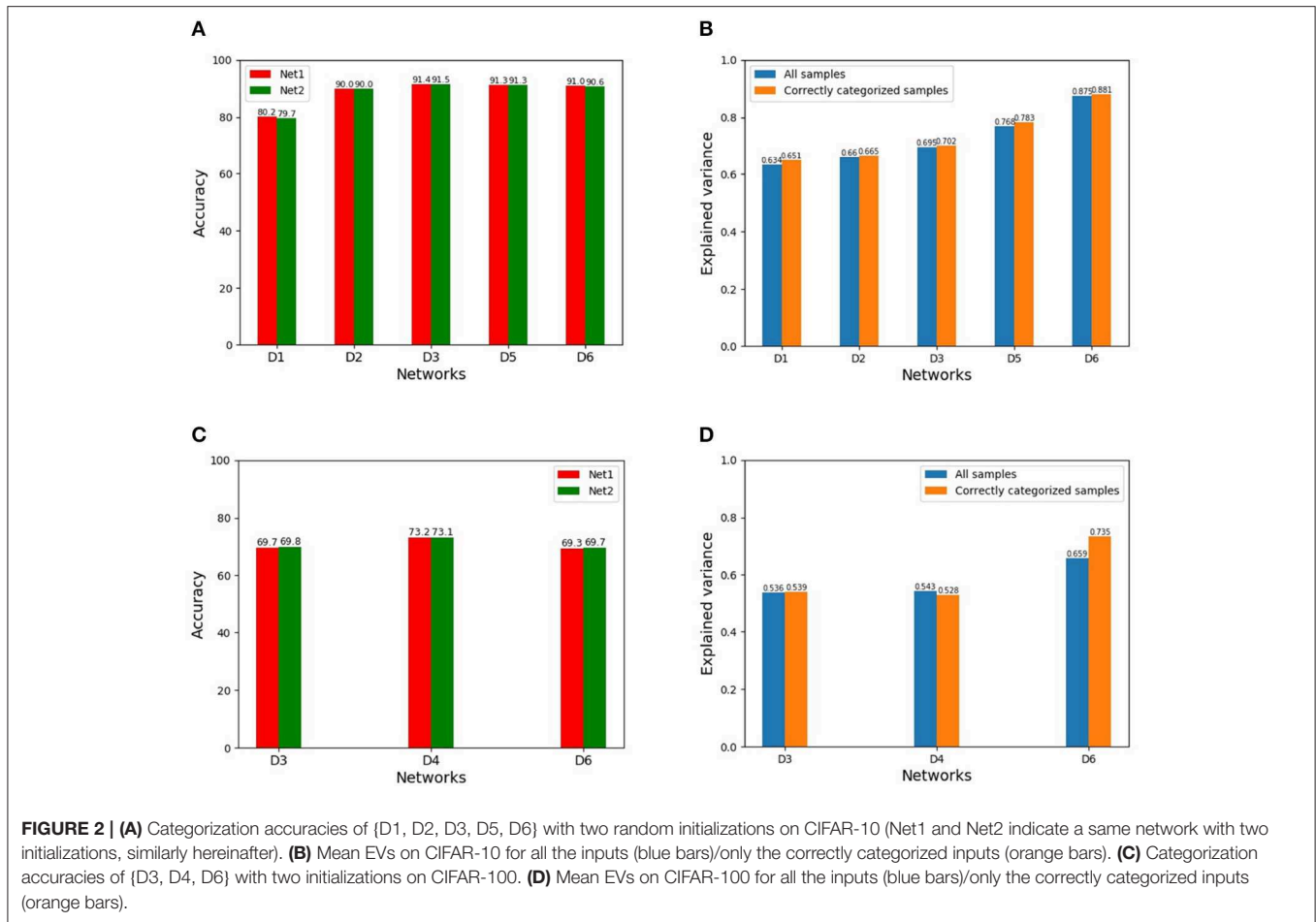
(ii) Explained variance on noisy data

In Szegedy et al. (2014), it is reported that DCNNs are sometimes sensitive to adversarial images, that is, images slightly corrupted with random noise, which do not pose any significant problem

for human perception, but dramatically alter the categorization performance of DCNNs. Here, we assessed the noise effects on the representation equivalence on CIFAR-10. The input images are normalized to the range [0, 1], and Gaussian noise with mean 0 and standard variance $\sigma = \{0.01, 0.02, 0.03, 0.04, 0.05, 0.07, 0.1\}$ are added into these images, respectively. **Figure 3A** shows the corresponding categorization accuracies of similar performing pairs of DCNNs under different architectures, while **Figure 3B** shows the corresponding mean EVs. We find that even under the noise level $\sigma = 0.1$, the explained variance does not change much, although the categorization accuracy decreases notably.

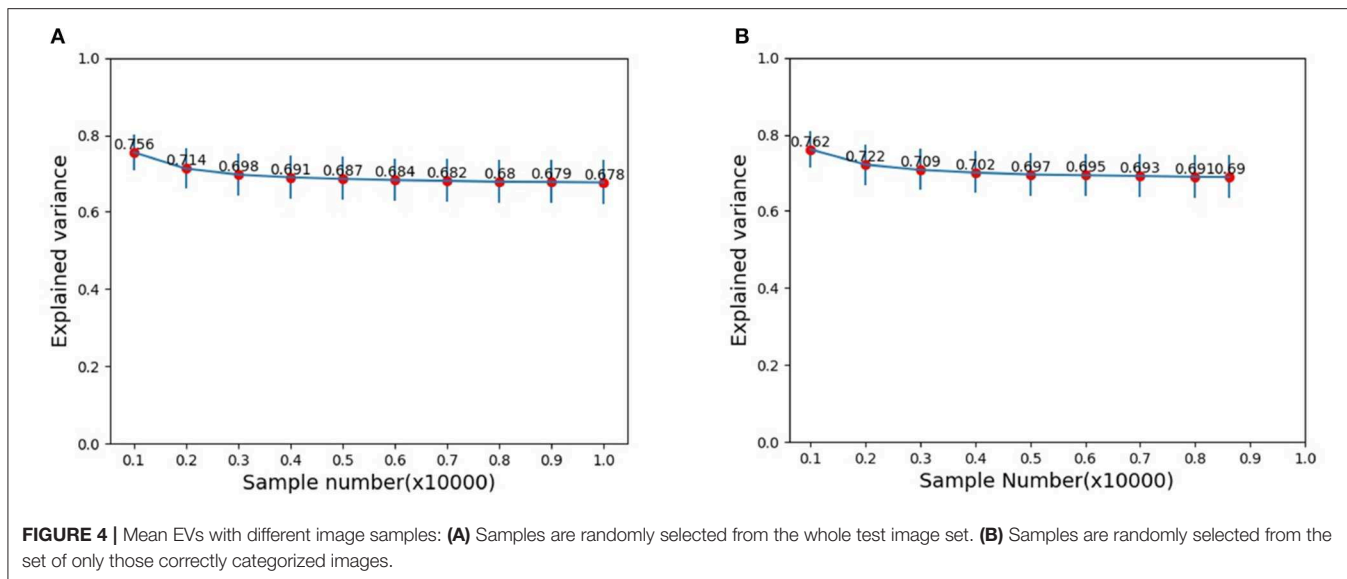
(iii) Variations of explained variance by changing stimuli size

In the neuroscience, the number of stimuli could not be too large. However, for image categorization by DCNNs, the size of the test set could be very large. Does the size of stimuli set play a role on the explained variance? To address this issue, we assessed the explained variance as the dataset size increases by resampling subsets from the original test set of images in CIFAR-10. Here, image subset sizes of [1000, 2000, ..., 10000] are evaluated. **Figures 4A,B** show the results on the resampled subsets from the whole set of test data and the set of only those images which are correctly categorized, respectively. Our results

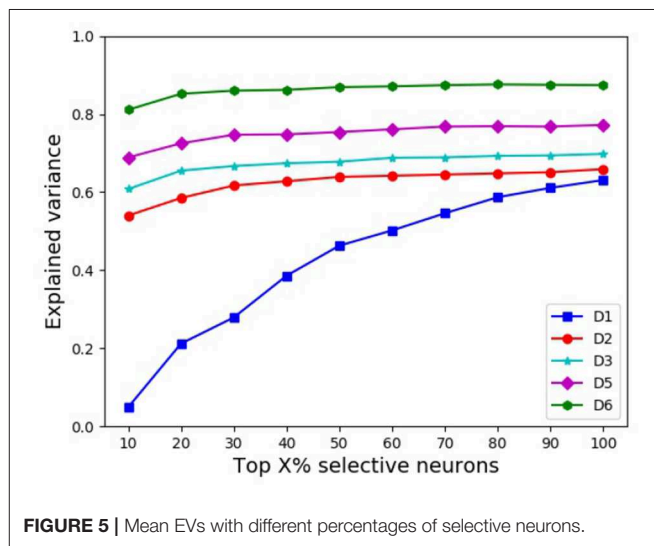


show that if the size of the stimuli set reaches a modestly large number (around 3000), the explained variance stabilized. That is to say, we do not need a too large number of stimuli for reliably

estimating explained variance. In other words, stimuli in the order of thousands could already reveal the essence, and a further increase of stimuli could not alter much the estimation.



(iv) Explained variance vs. neuron selectivity
 Clearly, some DCNN neurons are more selective than others (Dong et al., 2017, 2018). Using the kurtosis (Lehky et al., 2011) of the neuron’s response distribution to image stimuli, we investigated whether neuron selectivity has some correlation with the explained variance. We chose top {10%, 20%, . . . , 100%} most selective neurons from each DCNN in a similar performing pair, respectively, then computed the explained variance between the two chosen subsets, and the results are shown in **Figure 5**. As seen from **Figure 5**, with the increase of the percentage of selective neurons, the explained variance increases accordingly. This indicates that for the object representations of a similar performing pair of DCNNs, neuron selectivity is also an influential factor on their explained variance. The explained variance between the subsets of more selective neurons is smaller, and this result seems to be in concert with the conclusion in Morcos et al. (2018) where it is shown that neuron selectivity does not imply the importance in object generalization ability.



(v) A good representation does not necessarily needs IT-like
 In the literature (Khaligh and Kriegeskorte, 2014), it is shown that if an object representation is IT-like, it can give a good object recognition performance. This work shows that the inverse is not necessarily true, at least theoretically speaking. That is, as shown in the above experiments and discussions, many different representations can give the same or quite similar recognition results with/without noise.

Remark 6: In this work, we assume the final classifier is a Softmax classifier. For other linear classifiers, the general concluding remark of non-equivalence can be similarly derived. Of course, if the used classifier is a non-linear one, or the output of the penultimate layer is further processed by a non-linear operator before inputting it to a linear classifier, as done in Chang Tsao (2017), where a 3-order polynomial is used as a preprocessing step for the final classification, our results will no longer hold. But as shown in Majaj et al. (2015), monkey IT neuron responses can be reliably decoded by a linear classifier, we thought using Softmax as the final classifier for DCNN-based

IT cortex modeling could not constitute a major problem for our results.

3. CONCLUSION

Here, we would say that we are not against using DCNNs to model sensory cortex. In fact, its potential and usefulness have been demonstrated in Yamins et al. (2014) and Yamins and DiCarlo (2016). Here, we only provide a theoretical reminder on the possible non-uniqueness phenomenon of the learnt object representations by DCNNs, in particular, by the goal-driven approach proposed in Yamins and DiCarlo (2016). As shown in the convergent-learning literatures, such a non-uniqueness phenomenon is prevalent in deep learning, hence when DCNNs are used for modeling sensory cortex as a general framework,

people should be aware of this potential and inherent non-uniqueness problem, and appropriate network architectures in DCNN learning should be carefully considered.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: <http://www.cs.toronto.edu/~kriz/cifar.html>.

AUTHOR CONTRIBUTIONS

ZH conceived of the non-uniqueness phenomenon of object representation in modeling IT cortex by DCNN. QD and ZH explored the method. QD and BL implemented the explored

method and performed the validation. QD and ZH wrote the paper.

FUNDING

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB32070100), and National Natural Science Foundation of China (61991423, U1805264, 61573359, 61421004).

ACKNOWLEDGMENTS

This manuscript has been released as a Pre-Print at <http://export.arxiv.org/pdf/1906.02487> (Dong et al., 2019).

REFERENCES

- Cadieu, C., Hong, H., Yamins, D., Pinto, N., Ardila, D., Solomon, E., et al. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput. Biol.* 10:e1003963. doi: 10.1371/journal.pcbi.1003963
- Chang, L., and Tsao, D. (2017). The code for facial identity in the primate brain. *Cell* 169, 1013–1028. doi: 10.1016/j.cell.2017.05.011
- Csáji, B. C. (2001). *Approximation With Artificial Neural Networks*. Faculty of Sciences, Eötvös Loránd University, Hungary.
- Dong, Q., Liu, B., and Hu, Z. (2017). Comparison of IT neural response statistics by simulations. *Front. Comput. Neurosci.* 11:60. doi: 10.3389/fncom.2017.00060
- Dong, Q., Liu, B., and Hu, Z. (2019). Non-uniqueness phenomenon of object representation in modelling IT cortex by deep convolutional neural network (DCNN). *arXiv* 1906.02487.
- Dong, Q., Wang, H., and Hu, Z. (2018). Statistics of visual responses to image object stimuli from primate AIT neurons to DNN neurons. *Neural Comput.* 30, 447–476. doi: 10.1162/neco_a_01039
- Elston, G. N. (2002). Cortical heterogeneity: implications for visual processing and polysensory integration. *J. Neurocytol.* 31, 317–335. doi: 10.1023/A:1024182228103
- Elston, G. N. (2007). “Evolution of the pyramidal cell in primates,” in *Evolution of Nervous Systems, IV*, eds J. H. Kaas and T. D. Preuss (Oxford: Academic Press), 191–242.
- Elston, G. N., and Fujita, I. (2014). Pyramidal cell development: postnatal spinogenesis, dendritic growth, axon growth, and electrophysiology. *Front. Neuroanat.* 8:78. doi: 10.3389/fnana.2014.00078
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). “Deep residual learning for image recognition,” in *Proceedings of CVPR* (Las Vegas, NV).
- Hong, H., Yamins, D., Majaj, N., and DiCarlo, J. (2016). Explicit information for category-orthogonal object properties increases along the ventral stream. *Nat. Neurosci.* 19, 613–622. doi: 10.1038/nn.4247
- Jacobs, B., and Scheibel, A. B. (2002). “Regional dendritic variation in primate cortical pyramidal cells,” in *Cortical Areas: Unity And Diversity*, eds A. Schüz and R. Miller (London: Taylor and Francis), 111–131.
- Khaligh-Razavi, S., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* 10:e1003915. doi: 10.1371/journal.pcbi.1003915
- Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. Rev. Vis. Sci.* 1, 417–446. doi: 10.1146/annurev-vision-082114-035447
- Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*. Technique Report; Canada: University of Toronto.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “ImageNet classification with deep convolutional neural networks,” in *Proceedings of Advances in Neural Information Processing 25* (Lake Tahoe).
- Lehky, S., Kiani, R., Esteky, H., and Tanaka, K. (2011). Statistics of visual responses in primate inferotemporal cortex to object stimuli. *J. Neurophysiol.* 106, 1097–1117. doi: 10.1152/jn.00990.2010
- Li, Y., Yosinski, J., Clune, J., Lipson, H., and Hopcroft, J. (2016). “Convergent learning: do different neural networks learn the same representations?” in *Proceedings of ICLR* (San Juan, Puerto Rico).
- Luebke, J. I. (2017). Pyramidal neurons are not generalizable building blocks of cortical networks. *Front. Neuroanat.* 11:11. doi: 10.3389/fnana.2017.00011
- Majaj, N., Hong, H., Solomon, E., and DiCarlo, J. (2015). Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. *J. Neurosci.* 35, 13402–13418. doi: 10.1523/JNEUROSCI.5181-14.2015
- Morcos, A., Barrett, D., Rabinowitz, N., and Botvinick, M. (2018). “On the importance of single directions for generalization,” in *Proceedings of ICLR* (Vancouver, BC).
- Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv* 1409.1556.
- Spruston, N. (2008). Pyramidal neurons: dendritic structure and synaptic integration. *Nat. Rev. Neurosci.* 9, 206–221. doi: 10.1038/nrn2286
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). “Going deeper with convolutions,” in *Proceedings of CVPR* (Boston, MA), 1–9.
- Szegedy, C., Zaremba, W., Sutskever, Y., Bruna, J., Erhan, D., Goodfellow, I., et al. (2014). “Intriguing properties of neural networks,” in *Proceedings of ICLR* (Banff, AB).
- Tian, Y. (2017). “An analytical formula of population gradient for two-layered ReLU network and its applications in convergence and critical point analysis,” in *Proceedings of ICML* (Sydney, NSW), 3404–3413.
- Wang, L., Hu, L., Gu, J., Hu, Z., Wu, Y., He, K., et al. (2018). “Towards understanding learning representations: to what extent do different neural networks learn the same representation,” in *Proceedings of NIPS* (Montreal, QC), 9607–9616.
- Yamins, D., and DiCarlo, J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244
- Yamins, D., Hong, H., Cadieu, C., Solomon, E., Seibert, D., and DiCarlo, J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 111, 8619–8624. doi: 10.1073/pnas.1403112111

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Dong, Liu and Hu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

APPENDIX

Procedure to train DCNN₁ and DCNN₂:

Input: A set of n image objects: $D = \{I_i, i = 1, 2, \dots, n\}$ with known categorization labels.

Output: DCNN₁ and DCNN₂ whose object representations are different but with the same (or similar) categorization performance;

- 1 Using $D = \{I_i, i = 1, 2, \dots, n\}$ to train a DCNN by optimizing the categorization performance. This training can be done similarly as reported in numerous image categorization literatures. Denote the trained DCNN as DCNN₁. The output of the penultimate layer in DCNN₁ for D is denoted as $X = \{x_i, i = 1, 2, \dots, n\}$, x_i is the output for input image I_i . Denote the output of the final layer in DCNN₁ for D as: $X' = \{x'_i, i = 1, 2, \dots, n\}$, the weighting matrix at the final layer in DCNN₁ is W_1 and the bias vector is b_1 , that is $x'_i = W_1 x_i + b_1$;
- 2 Choose a non-linear monotonically increasing function $f(\cdot)$, and compute $Y' = \{y'_i, i = 1, 2, \dots, n\}$, where $y'_i = f(x'_i)$ in element-wise mapping;
- 3 Choose a weighting matrix W_2 for the second DCNN, say $W_2 = W_1$;
- 4 Compute $Y = \{y_i, i = 1, 2, \dots, n\}$ by $y_i = (W_2^T W_2)^+ W_2^T (y'_i - b_2)$;
- 5 Using training pair $\{(I_i \leftrightarrow y_i), i = 1, 2, \dots, n\}$ to train the second DCNN to minimize the Euclidean loss between the DCNN's output \tilde{y}_i and y_i .
- 6 The trained DCNN in step (5) is our required DCNN₂. The object representation x_i of DCNN₁ and y_i of DCNN₂ are different representations by Definition 2, because for the same object I_i , x_i and y_i can give the same categorization results in theory without noise, or similar results with noise in practice, but they cannot be transformed by a linear transformation as shown in Proposition 2.