NEURAL COLLOBRATIVE NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents a conceptually general and modularized neural collaborative network (NCN), which overcomes the limitations of the traditional convolutional neural networks (CNNs) in several aspects. Firstly, our NCN can directly handle non-Euclidean data without any pre-processing (e.g., graph normalizations) by defining a simple yet basic unit named neuron array for feature representation. Secondly, our NCN is capable of achieving both rotational equivariance and invariance properties via a simple yet powerful neuron collaboration mechanism, which imposes a "glocal" operation to capture both global and local information among neuron arrays within each layer. Thirdly, compared to the state-of-the-art networks that using large CNN kernels, our NCN with considerably fewer parameters can also achieve their strengths in feature learning by only exploiting highly efficient 1×1 convolution operations. Extensive experimental analyses on learning feature representation, handling novel viewpoints, and handling non-euclidean data demonstrate that our NCN can not only achieve state-of-the-art performance but also overcome the limitation of the conventional CNNs. [1].

1 INTRODUCTION

Artificial neural networks have been extensively studied and applied over the past three decades, achieving remarkable accomplishments in artificial intelligence. Among such networks, two types of neural networks have had a large impact on the research community. The first type is the multi-layer perceptron (MLP), which first became popular and effective via the development of the back-propagation training algorithm Rumelhart et al. (1986); Lecun (1987). However, because each neuron of the hidden layer is assigned with a private weight, the number of parameters inside MLP increases quite rapidly and easily leads to overfitting. Moreover, MLP has difficulty in representing the spatial structure of 2D data (e.g., images). The second type is convolutional neural networks (CNNs) LeCun et al. (1990). Motivated by the biological visual cortex model, CNNs propose to group adjacent neurons to share identical weights and represent 2D data by capturing the local pattern (i.e., receptive field) of each neuron.

Although CNNs have been proven to be significantly superior over MLP, they have several drawbacks, as highlighted by Sabour et al. (2017). Specifically, due to only abstracting information from local neighborhood pixels, the convolution operation inside each layer of CNNs ignores global information and fails to achieve rotational equivariance and invariance. Consequently, CNNs can not well handle image data with general rotational changes. Hence, unlike the human brain that does not depend on view angle to recognize objects, CNNs require a large amount of training data under various viewpoints as an important regularization for avoiding misrecognition. Moreover, unlike MLP, conventional CNNs cannot be directly applied for non-Euclidean data (e.g., graph data), which are quite common in the area of machine learning.

Attempting to address the aforementioned issues, several new networks Sabour et al. (2017); Hinton et al. (2018); Wang et al. (2017) have recently been proposed and achieved promising results. However, all of these methods have not been studied further or deeper to evolve CNNs, i.e., they still employ additional convolutional layers for feature learning. In contrast, this work focuses on developing a completely modularized network named neural collaborative network (NCN), which inherits the strengths of both MLP and CNNs and overcomes their drawbacks by introducing a simple

[1]The source codes will be released to facilite future researches after the review period for ensuring the anonymity

yet powerful neuron collaborative mechanism. Specifically, our NCN defines a simple yet basic unit named neuron array, which is a unit of vector in a meta data. As depicted in Fig. 1, a neuron array may be seen as a pixel of an image Fig. 1(a), a sampling of an audio Fig. 1(b), and a node of a general graph Fig. 1(c).

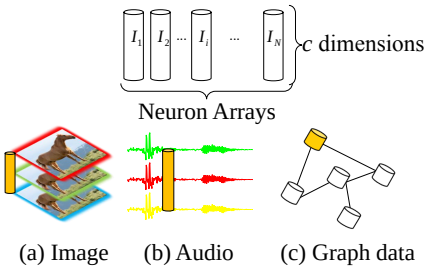

Figure 1: Neuron arrays are presented in form of orange cylinder in (a) an image, (b) an audio, and (c) a general graph.

Given the input neuron arrays, our proposed NCN enforces that they work collaboratively to represent each other according to our proposed neuron collaborative mechanism, which defines that the output for each neuron array is a linear combination of all the neuron arrays within this layer. Considering the limited compuation and memory resource, we developed a so-called "glocal" operation to efficiently obtain the collaborative representation. The advantages of our proposed NCN over the traditional convolutional networks are: i) *Rotational equivariance and invariance properties.* In contrast to convolution operations that use non-rotatable filters, the output of each layer inside our NCN is rotation-equivariant to the rotated input by enabling all the operations for the neuron arrays to have synchronous rotations. Thanks to the inherent equivariance property of our NCN, we can further achieve the invariance property via common global average pooling for classification-based tasks. ii) *Abstracting **glocal** information.* Our NCN considers the relationships among neuron arrays as well as their relative location information in a data-driven manner. In this way, our NCN is capable of capturing global and local (i.e., glocal) structural dependencies. iii) *Supporting non-Euclidean data.* Since our defined neuron arrays are basically vectors similar to MLP, both Euclidean data (e.g., image, video and voice) and non-Euclidean data can be used directly without pre-processing steps.

The **main** contributions of this paper are two-fold. Firstly, we propose a conceptually general yet powerful network, which is capable of generating powerful feature representations with rotational equivariance and invariance properties. Secondly, to the best of our knowledge, we are the first to design a completely modularized network, which overcomes the limitation of the existing CNNs while preserving their strength in feature learning. Extensive experiments demonstrate the superiority of our proposed NCN over the compared state-of-the-art network architectures (i.e., ResNet101 He et al. (2016b)) on the ImageNet benchmark Russakovsky et al. (2015), and our NCN can achieve comparable performance to the state-of-the-art methods on the CIFAR10 and Cora documentation classification benchmarks. Moreover, comprehensive experimental analyses validate the equivariance and invariance properties of our NCN.

2 Neural Collaborative Networks

In this section, we first present the neuron collaborative mechanism with detailed descriptions and comprehensive explanations. Then, we further theoretically prove that our NCN has an inherent rotational equivariance property and can obtain an invariance property via a simple step. Finally, we discuss our NCN with related works, and clarify its compatibility with existing CNN techniques or tricks.

Neuron Collaborative Mechanism. The motivation of this mechanism is to jointly leverage the collaborative representation among neurons. Specifically, suppose the input data (*e.g.*image, voice, graph matrix or their features) is fed into N neuron arrays $\mathbf{I} = \{I_i\}_{i=1}^N$, where I_i is a vector of c dimensions (see Fig. 1). Inspired by Cai et al. (2016); Zhang et al. (2011), we propose to obtain the output $\mathbf{y}_i$ for each neuron array I_i via a linear combination of all the neuron arrays $\mathbf{I}$ as follows:

$$\mathbf{y}_i = \mathbf{I}\mathbf{d}_i \tag{1}$$

where $\mathbf{D} = \{\mathbf{d}_i\}_{i=1}^N \in \mathbb{R}^{N\times N}$ denotes the collaborative matrix, which consists of relationships among the input neuron array set $\mathbf{I} = \{I_i\}_{i=1}^N$. For instance, given an image with n pixels, a "collaborative matrix" denotes an $n \times n$ matrix measuring the degree of relevance between any two pixels. In this work, we propose to calculate $\mathbf{D}$ by employing a completely learnable parameter $\omega^{\mathbf{D}}$ as follows:

$$\mathbf{D}_{ij} = f(\omega_{ij}^{\mathbf{D}} I_i); \quad s.t. \ \forall\, a, b,\ \omega_{ab}^{\mathbf{D}} = \omega_{ij}^{\mathbf{D}} \text{ if } |a-b| = |i-j|, \tag{2}$$

where i and j are the indexes that enumerate all items of $\mathbf{I}$. $f(\cdot)$ denotes a typical mentioned non-linear mapping. In our experiments, we choose the combination of LayerNorm Ba et al. (2016)+ReLU Krizhevsky et al. (2012) or BatchNorm Ioffe & Szegedy (2015)+ReLU Krizhevsky et al. (2012) for their widely application inside CNNs. To achieve rotational equivariance and invariance, we introduce an index-based constraint for the learnable parameter $\omega^{\mathbf{D}}$. In fact, irrespective of how the viewpoint of the input data changes, the relative spatial distance of its items remains unchanged. Therefore, we consider the distance in indexes as a hard constraint by enforcing that some neuron arrays with the same spatial distance share the same collaboration parameter. Specifically, for any four neuron arrays with indexes a, b, i and j, we enforce that their collaboration parameter $\omega^{\mathbf{D}}_{ab}$ equals $\omega^{\mathbf{D}}_{ij}$ when their relative distances are identical. In this work, we directly employ the l_1-norm, i.e., $|a-b| = |i-j|$, for ease of implementation. Note that, the distance is defined as the spatial distance between pixels for an image, while the distance denotes the Dijkstra shortest path distance between nodes for the graph data.

In addition to the implicit collaboration measurement for $\mathbf{D}$, our NCN has also considered the explicit collaboration measurement as Krähenbühl & Koltun (2011). Specifically, we additionally introduce another neuron array set $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^N$ with the parameter $\omega^{\mathbf{V}}$ to calculate a new collaborative matrix $\mathbf{A} = \{\mathbf{a}_i\}_{i=1}^N$. For each neuron array I_i, we first perform a simple neuron self-transformation (NST) as $\mathbf{v}_i = f(\omega^{\mathbf{v}} I_i)$, where $\omega^{\mathbf{v}} \in \mathbb{R}^{d \times c}$ is the parameter inside NST for transformation and $f(\cdot)$ implies aforementioned non-linear mapping. Then, we have:

$$\mathbf{A}_{ij} = f(\omega^{\mathbf{A}}_{ij}[-\|\mathbf{v}_i - \mathbf{v}_j\|^2; \mathbf{v}_i; \mathbf{v}_j]), \quad s.t. \ \forall \ a, b, \ \omega^{\mathbf{A}}_{ab} = \omega^{\mathbf{A}}_{ij} \text{ if } |a-b| = |i-j|, \tag{3}$$

where the optimization of $\omega^{\mathbf{A}}_{ij}$ is similar to that of $\omega^{\mathbf{D}}_{ij}$. Similar to $\mathbf{D}$, the collaborative representation $\mathbf{y}'_i$ for I_i by using $\mathbf{a}_i$ over the entire $\mathbf{I}$ is defined as: $\mathbf{y}'_i = \mathbf{I}\mathbf{a}_i$. Actually, the filter learned by our NCN is radial symmetry (also known as rotational symmetry). Therefore, the output feature map of these filters is ensured to be rotational invariant. This property distinguishes our NCN with Graph CNN Kipf & Welling (2017). More importantly, our NCN not only achieves rotational invariance properties but also enables powerful representation learning. This may imply that rotational-invariant filters are sufficient to learn strong models on complicated computer vision problems.

Actually, $\mathbf{D}$ and $\mathbf{A}$ are usually quite large under the practical use. Hence, directly calculating $\mathbf{D}$ is infeasible due to its memory and computation consuming. To address this issue and capture structural dependencies from both global and local perspectives, we design a "glocal" operation to efficiently calculate $\mathbf{D}$ and $\mathbf{A}$. We also define the upsampling and downsampling operations of neuron arrays, i.e., if the stride s does not equal 1, then we will achieve upsampled ($s < 1$) and downsampled ($s > 1$) results. Specifically, if $s < 1$, then we employ the bilinear interpolation strategy to increase the number of neuron arrays by generating a new neuron array from two adjacent neuron arrays. If $s > 1$, then we directly discard the neuron array for every $s - 1$. Specifically, we first downsample the input data using the downsampling operation to extract only few representative neuron arrays (*e.g.*16 (=4×4) representative neuron arrays). Then, we calculate their $\mathbf{D}$ and $\mathbf{A}$ and further directly interpolate them to the whole neuron arrays via a nearest-neighbor strategy. According to our experimental results, this step can well capture global information of neuron arrays and greatly improve efficiency. As a compensation, we impose a local region definition, which focuses on calculating the correlations among neighborhood neuron arrays to capture local structural information. Specifically, the i and j in Eq. (1) subject to $dis(i,j) < \varepsilon$, where $dis(\cdot)$ denotes the shortest path distance between i and j, and ε is a threshold (*e.g.*$\varepsilon = 16$). In this way, we can obtain local collaborative representation $\mathbf{y}''_i$ and $\mathbf{y}'''_i$.

Finally, our proposed NCN directly fuses these two types of collaborative representations to obtain the final output $\mathbf{z}_i$ for I_i as follows:

$$\mathbf{z}_i = f(\omega^{\mathbf{z}}[\mathbf{y}_i; \mathbf{y}'_i; \mathbf{y}''_i; \mathbf{y}'''_i]), \tag{4}$$

where $[\cdot;\cdot;\cdot;\cdot]$ denotes the concatenate operation and $\omega^{\mathbf{z}}$ represents the parameters for the fusion. With this simple yet powerful neuron collaborative mechanism, we can design any types of NCNs.

Rotational Equivariance Property. Our proposed NCN can always achieve rotational equivariance. We present our clarifications as follows. Suppose that we have input image data with two arbitrary pixels, i.e., i and j with coordinates (x, y) and (u, q), respectively. After rotating θ degrees, the new coordinates for i and j are $T(i,\theta) = (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$ and $T(j,\theta) = (u\cos\theta - q\sin\theta, u\sin\theta + q\cos\theta)$, respectively. T denotes the rotation function. Meanwhile, we have the following equation:

$$I_i = I_{T(i,\theta)}; \ \ I_j = I_{T(j,\theta)}. \tag{5}$$

To demonstrate the rotational invariance of our NCN (denoted as the function $g(\cdot)$), our objective is to prove that $g(I_{T(i,\theta)}) = T(g(I_i), \theta)$, i.e., the output of our NCN for the rotated input should be identical to rotating the output of the original input by the same angle.

We have the following equations via Eq. (4):

$$g(I_{T(i,\theta)}) = f(\omega^{\mathbf{z}}[\mathbf{y}_{T(i,\theta)}; \mathbf{y}'_{T(i,\theta)}]). \tag{6}$$

For clarity, we only consider $\mathbf{y}'_{T(i,\theta)}$. A similar derivation procedure can also be applied to $\mathbf{y}_{T(i,\theta)}$ without difficulty. Then, we have

$$g(I_{T(i,\theta)}) = f(\omega^{\mathbf{z}} T(\mathbf{X}, \theta)\mathbf{a}_i); T(g(I_i), \theta) = T(f(\omega^{\mathbf{z}}\mathbf{X}\mathbf{a}_i), \theta) = f(\omega^{\mathbf{z}} T(\mathbf{X}, \theta)\mathbf{a}_{T(i,\theta)}) \tag{7}$$

where the derivation is based on the fact that $\omega^{\mathbf{z}}$ are shared among these neuron arrays irrelevant to i. Therefore, we only need to prove that $\mathbf{a}_i = \mathbf{a}_{T(i,\theta)}$, i.e., $\mathbf{A}_{ij} = \mathbf{A}_{T(i,\theta)T(j,\theta)}$. According to Eq. (3), we denote $\mathbf{A}_{ij}$ as $h(\|\mathbf{v}_i - \mathbf{v}_j\|^2)$. In the following, we prove that $\|\mathbf{v}_{T(i,\theta)} - \mathbf{v}_{T(j,\theta)}\|^2 = \|\mathbf{v}_i - \mathbf{v}_j\|^2$ detailed in the supplementary material. In this way, we can obtain $g(I_{T(i,\theta)}) = T(g(I_i), \theta)$, which is occasionally called *rotational equivariance* and is quite beneficial for many types of tasks, such as document classification and semantic image segmentation.

Rotational Invariance Property. For the general classification task, it is desirable to learn a representation that is invariant to any viewpoint, i.e., rotational invariance. To achieve this goal, we propose simply performing a widely used yet efficient operation, i.e., global average pooling, on the rotational equivariance features obtained above. Next we prove this invariance property. Specifically, our goal is to prove that $\mathcal{S}(I) = \mathcal{S}(T(I, \theta))$, where the function $\mathcal{S}(\cdot)$ denotes the final feature representation of our NCN. By incorporating the global average pooling, we have the following:

$$\mathcal{S}(I) = \frac{1}{N}\sum_{i=1}^{N} g(I_i); \mathcal{S}(T(I,\theta)) = \frac{1}{N}\sum_{i=1}^{N} g(I_{T(i,\theta)}) = \frac{1}{N}\sum_{i=1}^{N} T(g(I_i), \theta) = \frac{1}{N} T(\sum_{i=1}^{N} g(I_i), \theta),$$

where the derivation is based on the rotational equivariance obtained above. Because $\sum_{i=1}^{N} g(I_i)$ denotes an image with a single pixel, $T(\sum_{i=1}^{N} g(I_i), \theta) = \sum_{i=1}^{N} g(I_i)$. Hence, $\mathcal{S}(I) = \mathcal{S}(T(I, \theta))$, which proves that the rotation cannot change the feature representation of our NCN.

Since all of the aforementioned formulations are fully differentiable, we can directly exploit the standard backpropagation algorithm with stochastic gradient descent (SGD) LeCun et al. (1990) to optimize all the parameters in the training phase. For the testing, we also directly perform the forward propagation for obtaining the final output $\mathbf{z}_i$ for each input I_i. During training, we optimize $\omega^{\mathbf{D}}$ via $\omega^{\mathbf{D}}_{ij} := \omega^{\mathbf{D}}_{ij} + lr \sum_{\forall a,b,|a-b|=|i-j|} \nabla\omega^{\mathbf{D}}_{ab}$, where lr denotes the learning rate.

Comparison with Non-local Networks (NonlocalNet). NCN advances NonlocalNet in three aspects. First, the NonlocalNet used non-local operations among pixels to capture long-range dependencies within feature maps. Differently, our NCN employs a new neural collaboration mechanism to handle both global and local information of input data. Our proposed mechanism is capable of explicitly modeling the structural correlation within images. The weak accuracy of NonlocalNet in the following experiment section shows that NonlocalNet still relies on the convolutional operations for capturing structural and contextual dependencies, proving that it also suffers from the limitations of the conventional CNNs. Second, different from NCN, NonlocalNet cannot enable rotation-invariant representation learning. Last, even the global operation in NCN is different from NonlocalNet. As described above, our global operation is built upon the patches (using downsampling operation). This enables NCN to well capture global information and greatly improve efficiency. The experimental results also verify the superior of NCN over NonlocalNet.

Compatibility with CNN Tricks and Techniques. Our proposed NCN is quite compatible with most existing tricks and techniques for CNNs. For instance, through embedding a batch normalization Ioffe & Szegedy (2015) layer into every non-linear mapping function $f(\cdot)$, our NCN can support a large learning rate for high learning efficiency. Meanwhile, we can also exploit the residual connection strategy He et al. (2016b) to create a short-cut inside our NCN.

3 EXPERIMENTS

3.1 LEARNING FEATURE REPRESENTATION (IMAGENET)

	Top-1 Acc. (%)	Top-5 Acc. (%)	# Params
ResNet101	77.4	93.7	84.45M
NCN	**79.2**	**94.3**	**67.11M**

Table 1: ImageNet *val* top-1 and top-5 accuracies.

We have compared our NCN with a quite representative and cutting-edge network, i.e., ResNet101 He et al. (2016a), on the ImageNet 2012 classification benchmark Russakovsky et al. (2015), which contains 1.28M images with $1,000$ categories for training. The bottleneck of ResNet101 has three layers, each of which includes convolution, ReLU and batch normalization (BN) Ioffe & Szegedy (2015), to abstract informative patterns from data. Similarly, our proposed NCN always consists of "three layers": two to calculate the collaborative representation $\{\mathbf{Y}, \mathbf{Y}', \mathbf{Y}', \mathbf{Y}'''\}$ and the other one to obtain their fusion $\mathbf{Z}$. Therefore we directly replace all the residual bottlenecks inside ResNet101 with our method with the same dimensions(*i.e.*channels in ResNet) and feature map size, we conduct the comparison under the same single-crop & single-scale evaluation settings as He et al. (2016a), which is the current standard criterion used by Szegedy et al. (2015); Dai et al. (2017).

Tab. 1 reports the top-1/top-5 accuracies of our NCN and ResNet101 on the 50k validation images. As shown in Tab. 1, our NCN surpasses ResNet101 by a clear margin in terms of both the top-1 and top-5 accuracies. In particular, our NCN performs approximately 2% better than ResNet101 on the top-1 accuracy (79.2% *vs.* 77.4%) thanks to our proposed neural collaboration mechanism. Note that, our reproduced top-1 accuracy (77.4%) for ResNet101 actually equal to the official result (Caffe implementation: 76.4%; Tensorflow: 77.4%). This result demonstrates that our proposed NCN can learn more discriminative representations than ResNet101. Meanwhile, Tab. 1 clearly shows that the model complexity of our NCN is significantly lower (approximately 21% less) than ResNet101 (67.11M *vs.* 84.45M) even though they both have the same network depths. The reason for this result is that most of the parameters inside the bottleneck of our NCN are from the NST function, which is 9 times more efficient than the 3×3 convolution inside the bottleneck of ResNet. In summary, these results demonstrate the effectiveness and efficiency of our proposed NCN.

3.2 Handling Novel Viewpoints

To verify the superiority of our NCN for handling novel viewpoints, we have conducted experiments on the CIFAR10 Krizhevsky & Hinton (2009) and smallNORB LeCun et al. (2004) benchmarks.

CIFAR10: consists of 50k training images and 10k testing images for 10 classes. According to the novel viewpoint setting, we employ a limited range of viewpoints for training, and we further test on a completely different range of viewpoints. As we know all the categories of objects (e.g., truck, horse, ship, and so forth) in CIFAR10 can be regarded as never upside-down. Intuitively, the images from both the original training and testing sets of CIFAR10 are all in the "normal" viewpoints, which are consistent with human perception or experience. To evaluate the rotational equivariance and invariance capacity of our proposed NCN, we simulate a new *test* set for the CIFAR10 benchmark by using the "novel" viewpoints. Specifically, all images are flipped vertically from the original *test* set.

Implementation Details: The detailed architecture of our NCN for this benchmark is summarized in Tab. 2. Since the inputs of our NCN are 32×32 images with per-pixel mean subtracted, we first exploit 5 NCN layers with 32×32 neuron arrays in 16 dimensions to capture its structural information. Then, we use the above-mentioned downsampling strategy to reduce the number of neuron arrays from 32×32 to 16×16 and feed the reduced neuron arrays into 5 NCN layers with 16×16 neuron arrays in 32 dimensions. Similarly, we further downsample the neuron arrays from 16×16 to 8×8 and feed them into 5 NCN layers with 8×8 in 64 dimensions. Note that we gradually increase the dimension of the neuron array inside the NCN layer to capture richer and more abstract information for retaining a discriminative representation. Finally, we build a softmax layer upon the last NCN layer to complete the final recognition task.

	Layers 1–5 (NCN Layer)	Layers 6–10 (NCN Layer)	Layers 11–15 (NCN Layer)	Layer 16 (Softmax)
Number	32×32	16×16	8×8	-
Dimensions	16	32	64	-

Table 2: Detailed architecture of our NCN used on the CIFAR10 benchmark. Note that "Number" denotes the number of neuron arrays within each layer, while "Dimensions" denotes the dimension of each neuron array.

Tab. 8 presents the accuracy comparison of both normal views and novel views for all the compared methods. In terms of normal viewpoints, it is clear that our NCN consistently outperforms ResNet (93.6% *vs.* 92.4%) and obtains superior accuracy over all the competing

methods. Importantly, as for novel viewpoints, our NCN achieves an incredible accuracy of 93.6%, which is exactly the same as the accuracy for normal viewpoints (i.e., 'Acc. Retain' equals 100% in Tab. 8). However, the performance of ResNet decreases from 92.44% to 43.80% due to the above-mentioned drawback of CNNs. This profound comparison clearly justifies that our NCN is a fundamental solution to address the issue of novel viewpoints.

Moreover, Tab. 8 also presents the model complexity comparison between our NCN and ResNet on the CIFAR10 benchmark. As shown in Tab. 8, by using only 47.8% of the parameters of ResNet, our NCN performs approximately 1.2% better than ResNet on the normal viewpoints and 53.1% better than ResNet on the novel viewpoints.This result further demonstrates the effectiveness and lightweight property of our proposed NCN.

Method	Novel	Normal
Maxout	-	90.62%
MIN	-	91.81%
DSN	-	91.78%
Highway	-	91.20%
Dynamic-Capsule	41.11%	89.4%
Matrix-Capsule	-	88.1%
ResNet-20	42.8%	91.8%
ResNet-32	40.5%	92.4%
ResNet-182	42.3%	94.4%
NCN-32	93.6%	93.6%
NCN-182	**95.0%**	**95.0%**

Table 3: Comparisons of the expanded CIFAR10 accuracy between ResNet and NCN on both "normal" and "novel" viewpoints, where normal indicates that the model is evaluated on the original *test* set, while novel indicates that the model is evaluated on our simulated *test* set. 'Acc. Retain' denotes the rate of novel and normal view accuracy. (a) CIFAR10 accuracy: the results are from **OFFICIAL** reports are in blue.

SmallNORB: Actually, learning viewpoint invariant knowledge is crucial in visual representation, and this issue was also addressed by the recently proposed Capsules Sabour et al. (2017); Hinton et al. (2018). Next we compare with our method with Capsules to evaluate NCN's ability on learning viewpoint invariant knowledge. We use the same benchmark of handling 3D viewpoint change (smallNORB LeCun et al. (2004)) as Capsules did. All methods are trained on one-third of the training data containing azimuth viewpoints of (300, 320, 340, 0, 20, 40) and tested on the two-thirds of the test data that contained azimuth viewpoints from 60 to 280. Note that there's no azimuth viewpoint overlap between training and testing. Results in Table 4 show that compared with the baseline CNN and Capsules, our NCN reduce the test error rate on novel viewpoints by about 45% and 18.5%, respectively, verifying the effectiveness of our NCN in handling novel viewpoints.

3.3 Evaluations on Non-Euclidean Data (Cora)

Networks	CNN	Dynamic-Capsule	Matrix-Capsule	**NCN**
Novel View Test	Novel = 20% Normal = 3.7%	Novel = 20% Normal = 3.7%	Novel = 13.5%[15], Normal = 3.7%[15]	Novel = **11.0%** Normal = 3.7%
Normal View Test	2.56%	2.7%	1.4%[15]	**1.38%**

Table 4: Comparisons among CNN, Capsules and NCN on smallNORB.

A common form of graph-structured data is a network of documents. For example, scientific documents in a database are related to each other through citations and references. Each document is a vertex in the graph with certain features, and a citation is an edge from one vertex to the other. Administrators of such large networks may desire to automatically label documents according to their relationships to the remainder of the literature. To demonstrate the compatibility of our NCN for non-Euclidean data, we adapt our proposed NCN to tackle such a vertex classification task on the Cora benchmark Sen et al. (2008), which is a large network of scientific publications connected through citations.

The vertex features in this case are binary word vectors that indicate the presence of a word from a dictionary of 1,433 unique words. There are 2708 publications classified under 7 different categories - case based, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning and theory. There is an edge connection from a cited article to a citing article and another edge connection from a citing article to a cited article. These edge features are also binary representations. We use a quite simple architecture: three NCN layers with outputs of 1,000 dimensions, followed by a classifier layer with outputs of 7 dimensions. The last classifier layer computes the prediction of each vertex. As for the Cora dataset is quite small, we perform 10-fold cross validations to form the training and test set for a fair comparison as the majority of methods Belkin et al. (2006); Kipf & Welling (2017).

Method	Accuracy (%)
ManiReg	59.5
SemiEmb	59.0
LP	68.0
DeepWalk	67.2
ICA	75.1
Planetoid*	75.7
Graph-CNN	81.5
MoNet	81.7
GAT	**83.0**
NCN	81.9

Table 5: Comparison with state-of-the-art on Cora document classification.

Comparisons with State-of-the-art Methods. In Tab. 5, we present the comparison with the current state-of-the-art approaches. It is clear that our NCN (81.9%) achieves comparable performance to the best of all competitive methods, e.g., Graph-CNN Kipf & Welling (2017) (81.5%). This comparison once again verifies the effectiveness of our NCN.

Neuron Collaboration *vs.* **Non-Collaboration**. We first evaluate the effectiveness of neural collaboration by constructing two different networks, *i.e.*our NCN and neural non-collaborative net. The non-collaborative net is obtained by directly replacing each NCN layer with a simple non-linear NST.

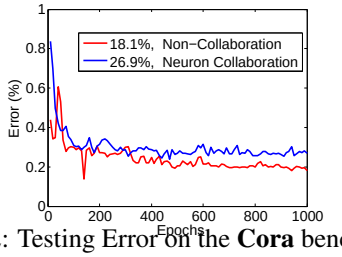

Figure 2: Testing Error on the **Cora** benchmark.

As shown in Fig. 2, our NCN consistently reaches substantially lower testing errors than the compared non-collaborative net during the entire training phase. Finally, our NCN obtains approximately 8% lower testing error rate than our NCN (26.9% *vs.* 18.1%). This difference demonstrates that the learning ability of our NCN is much better than that of neural non-collaborative net. Actually, neural non-collaborative net only performs NST for each neuron array, i.e., message communication between any two neurons is blocked within the same layer. This leads to neglecting some critical information (e.g., citation information in this task). In summary, our NCN fully exploits all the information among neurons, and performs substantially better than neural non-collaborative net. This comparison verifies the contribution of our neural collaboration mechanism.

Glocal *vs.* **Non-local**. In the previous experiments, we proved that the collaborative representation among neurons is beneficial for training. Now, we concentrate on investigating the structure manner of collaboration. In Tab. 6, we compare two operations: (A) Glocal operation. Structure messages (i.e., graph adjacency matrix) are explicitly incorporated in the collaborative representation, and (B) Non-local operation. No structure information is considered in our neural collaboration mechanism. This comparison is consistent with Section 2.

Method	Test Acc. (%)	Train Acc. (%)
Glocal	**81.9**	**99.9999**
Non-local	75.6	99.9982

Table 6: Analysis of glocal operation on Cora document classification.

Tab. 6 shows that glocal operation inherent in NCN is considerably better than the non-local operation (81.9% *vs.* 75.6%). We argue that this result is because the structure of the graph has been isolated in non-local operation. In particular, if we randomly shuffle the nodes in the graph, there will be no impact on the accuracy of non-local operation, which contradicts the fact that swapping nodes of a graph generally changes the property of a graph. Isolating the graph structure leads to suboptimal performance, which suggests the reasonableness of our used glocal operation.

3.4 Discovering Neuron Relationships

After training, we exhaustively forward-propagate all images of the CIFAR10 dataset to obtain their collaborative matrices of the neuron arrays, which are from the last NCN layer. Two motivating examples from the normal and novel views are depicted in Fig. 3. As shown, there are 8×8 collaborative matrices for 8×8 neuron arrays. Each collaborative matrix is also 8×8, which implies the relationship of its corresponding neuron array and all the 8×8 neuron arrays (including itself). For instance, the patch in the gray rectangle can be directly obtained by a weighted combination of all image patches with their correlations to that patch as weights. As shown in Fig. 3(a), it is clear that those neuron arrays with high correlations to the 18-th neuron array are exactly corresponding to the patches that belong to the horse category. As for Fig. 3(b), its 8×8 collaborative matrices are exactly the same as that of Fig. 3(a) after rotating by 180 degrees. This result further justifies the rotational equivariance property of our proposed NCN.

3.5 Comparison with Non-local Neural Networks

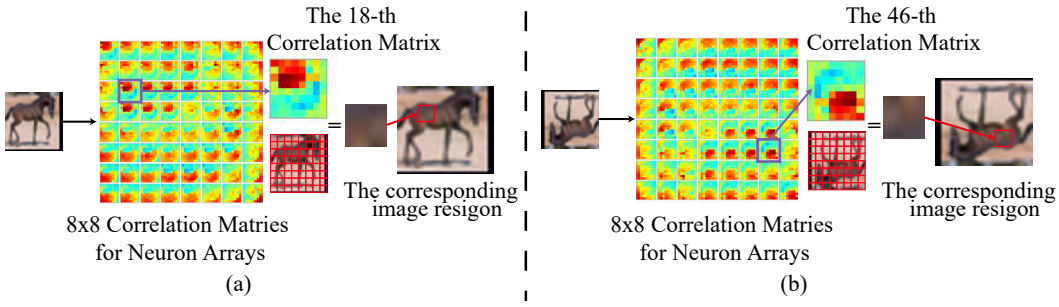

Figure 3: Analysis on correlations of the neuron arrays, which are from the last NCN layer.

We verify the superior of NCN over non-local neural networks (NonlocalNet) in three different aspects. First, we show whether NonlocalNet can handle viewpoint changes. Specifically, we insert NonlocalNet into every convolutional layer of ResNet (whose architecture is consistent with Tab. 2), denoted as ResNet+NonlocalNet. Table 7(a) is a comparison on handling the new object view on CIFAR10. The above results illustrate that the performance gain by ResNet+NonlocalNet is marginal and thus NonlocalNet cannot handle the viewpoint changes.

(a) Handing Viewpoint Change

Networks	ResNet	ResNet+NonlocalNet	**NCN**
Acc.	40.5%	40.9%	**93.6%**

(b) Standard Image Classification

Networks	ResNet	ResNet+NonlocalNet	**NCN**
Acc.	92.4%	74.2%	**93.6%**

(c) Efficiency Analysis

Networks	ResNet	**NCN**
Training Time	13.7	**20.6**
Testing Time	4.9	**6.9**
Memory (GB)	8.5	**11**
MACC (GFLOPS)	7.6	**11.2**

Table 7: Analysis on CIFAR10.

In another experiment of standard image classification, we show whether NonlocalNet can be modularized in representation learning. Specifically, we replace the convolutional operations in ResNet by our NST (neuron self-transformation operation), denoted as NonlocalNet+NST. The results on CIFAR10 are reported in Table 7(b). The weak accuracy of NonlocalNet+NST shows that NonlocalNet still relies on the convolutional operations for capturing structural and contextual dependencies, although it can handle global information of input data. In contrast, our NCN can model both local and global structural information in an explicit way, achieving the best performance.

3.6 Efficiency Analysis

We use a desktop with 8 Titan Xp GPUs to perform the training and testing. The time costs in millisecond per image per GPU are reported in Table 7(c). As shown, CNN is 33% and 42% faster than our NCN in the training and testing phase, respectively. The FLOPs comparison of MACC demonstrates that our NCN can achieve new state-of-the-art performance by requiring only about 47% more operations. Actually, this additional operations is acceptable.

4 Related Work

Evolutions of Convolutional Neural Networks: Although significant progress has been achieved in the architecture design of CNNs from LeNet LeCun et al. (1998) to more recent deep and powerful networks (e.g., ResNet He et al. (2016b)), evolving the structure of CNNs to overcome their drawbacks is also quite crucial and a long-standing problem in machine learning (*e.g.*Li et al. (2017)). This issue motivates many researchers to extend CNNs to include rotational equivariance Oyallon & Mallat (2015); Cohen & Welling (2016); Dieleman et al. (2016); Dai et al. (2017); Worrall et al. (2017); Henriques & Vedaldi (2017); Cohen et al. (2017); Cohen & Welling (2017); Wang et al. (2017); Cohen et al. (2018). Specifically, Dai *et al.* Dai et al. (2017) proposed to enhance the transformation modeling capability of CNNs by introducing learnable offsets to augment the spatial sampling locations within the feature map. Worrall *et al.* Worrall et al. (2017) presented harmonic networks to obtain rotation-invariant feature maps by returning the maximal response and orientation of circular harmonic filters. Sabour *et al.* Sabour et al. (2017) proposed employing a group of neurons

named a capsule to represent the instantiation parameters of a specific type of entity, such as an object and an object part. Building upon the work of Sabour et al. (2017), Hinton *et al.* Hinton et al. (2018) further presented a new type of capsule that has a logistic unit to represent the presence of an entity and a 4×4 pose matrix to represent the pose o f that entity. Motivated by the self-attention mechanism Vaswani et al. (2017), Wang *et al.* Wang et al. (2017) incorporated non-local operations into CNNs as a generic family of building blocks for capturing long-range dependencies. Although these methods achieved promising results, they still require the use of several convolutional layers for feature representation. Moreover, several limited attempts Kipf & Welling (2017); Such et al. (2017) have been made to extend CNNs for handling graph data. For instance, Kipf *et al.* Kipf & Welling (2017) presented a layer-wise propagation rule for CNNs to operate directly on graph-structured data. Such *et al.* Such et al. (2017) defined filters as polynomials of functions of the graph adjacency matrix for unstructured graph data. However, these variants of CNNs cannot use a unified framework to support both signals/images and graphs.

Actually, NCN advances them in four aspects. First, as R2 pointed out, they have "weaker empirical performance" than NCN. Second, their computational cost is high, *e.g.*Defferrard et al. (2016) is $7\times$ slower than the compared CNN. Third, a graph normalization is required to pre-process their input data, while the ambiguity problem of the graph norm remains unsolved. Fourth, the learned filters of these methods usually depend on a specific graph structure, leading to their limitations of generalization. Our NCN overcomes these limitations and beyond, thanks to the proposed neuron collaboration mechanism with the glocal operation.

Collaborative Representation: The collaborative representation mechanism corresponds to representing a test sample as a sparse linear combination of all training samples, and it has successfully been applied in sparse-representation-based methods Wright et al. (2009); Zhang et al. (2011); Cai et al. (2016) for visual recognition and in non-local techniques Buades et al. (2005); Protter et al. (2009) for image restoration. For instance, Zhang *et al.* Zhang et al. (2011) proposed an effective collaborative-representation-based classifier by utilizing the l_2-norm regularizer for the collaborative representation of a test sample from the training samples across all classes. Considering an image patch as a sample, Buades *et al.* Buades et al. (2005) proposed reconstructing a given local patch for image restoration via the collaborative representation of non-local similar patches, which are similar to this local patch and collected throughout the entire image. Inspired by these methods, we propose modeling the collaborative representation of neuron arrays to replace the conventional convolutional filtering. In contrast to these method that their defined collaborative mechanism ignore the local structural information, our proposed NCN consider the global and local structural information, simultaneously.

5 Conclusion

This paper presented a concise NCN to be a promissing substitute for overcoming the limitations of widely used deep CNNs without losing their strengths in feature learning. Specifically, our NCN advances in naturally considering the relationships among neurons to obtain non-local representations with equivariance and invariance properties. We further applied our proposed NCN for the recognition tasks of both Euclidean data and non-Euclidean data. Extensive experimental analyses from a variety of aspects justify the superiority of our NCN. In the future, we will extend our work to be suitable for more general tasks to demonstrate its superiority.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *arXiv:1607.06450 [stat.ML]*, 2016.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.

Sijia Cai, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. A probabilistic collaborative representation based approach for pattern classification. In *CVPR*, 2016.

Taco S. Cohen and Max Welling. Steerable cnns. In *ICLR*, 2017.

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Convolutional networks for spherical signals. In *arXiv:1709.04893 [cs.LG]*, 2017.

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *ICLR*, 2018.

T.S. Cohen and M. Welling. Group equivariant convolutional networks. In *ICML*, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.

Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *ICML*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016b.

Joao F. Henriques and Andrea Vedaldi. Warped convolutions: Efficient invariance to spatial transformations. In *ICML*, 2017.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *ICLR*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456, 2015.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 11(6):2278–2324, 1998.

Yann Lecun. *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*. Universite P. et M. Curie (Paris 6), 6 1987.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pp. 396–404, 1990.

Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pp. II–104. IEEE, 2004.

Xilai Li, Tianfu Wu, Xi Song, and Hamid Krim. Aognets: Deep and-or grammar networks for visual recognition. *arXiv preprint arXiv:1711.05847*, 2017.

Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *CVPR*, 2015.

M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE Trans. Image Processing*, 18(1):36–51, 2009.

D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by backpropagating errors. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1986.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Sara Sabour, Nicholas Frosst, and Geoffrey Hinton. Dynamic routing between capsules. In *NIPS*, 2017.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

Felipe Petroski Such, Shagan Sah, Miguel Domínguez, Suhas Pillai, Chao Zhang, Andrew Michael, Nathan D. Cahill, and Raymond W. Ptucha. Robust spatial filtering with graph convolutional neural networks. *J. Sel. Topics Signal Processing*, 11(6):884–896, 2017.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pp. 1–9, 2015.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *arXiv:1711.07971 [cs.LG]*, 2017.

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *CVPR*, 2017.

J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(2):210–227, 2009.

Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *ICCV*, 2011.

6 APPENDIX

6.1 ROTATIONAL EQUIVARIANCE PROPERTY

Our proposed NCN can always achieve rotational equivariance. We present our clarifications as follows. Suppose that we have input image data with two arbitrary pixels, i.e., i and j with coordinates (x, y) and (u, q), respectively. After rotating θ degrees, the new coordinates for i and j are $T(i, \theta) = (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$ and $T(j, \theta) = (u\cos\theta - q\sin\theta, u\sin\theta + q\cos\theta)$, respectively. T denotes the rotation function. Meanwhile, we have the following equation:

$$\begin{aligned} I_i &= I_{T(i,\theta)} \\ I_j &= I_{T(j,\theta)}. \end{aligned} \tag{8}$$

To demonstrate the rotational invariance of our NCL (denoted as the function $g(\cdot)$), our objective is to prove that $g(I_{T(i,\theta)}) = T(g(I_i), \theta)$, i.e., the output of NCL for the rotated input should be identical to rotating the output of NCL for the original input by the same angle.

We have the following equations via Eq. (4)@paper:

$$g(I_{T(i,\theta)}) = f([\mathbf{y}_{T(i,\theta)}; \mathbf{y}'_{T(i,\theta)}]; \omega^{\mathbf{z}}) \tag{9}$$

For clarity, we only consider $\mathbf{y}'_{T(i,\theta)}$. A similar derivation procedure can also be applied to $\mathbf{y}_{T(i,\theta)}$ without difficulty. Then, we have

$$\begin{aligned} g(I_{T(i,\theta)}) &= f(T(\mathbf{X}, \theta)\mathbf{a}_i; \omega^{\mathbf{z}}) \\ T(g(I_i), \theta) &= T(f(\mathbf{X}\mathbf{a}_i; \omega^{\mathbf{z}}), \theta) \\ &= f(T(\mathbf{X}, \theta)\mathbf{a}_{T(i,\theta)}; \omega^{\mathbf{z}}) \end{aligned} \tag{10}$$

where the third line is derived from the second line based on the fact that $\omega^{\mathbf{z}}$ are shared among these neuron arrays irrelevant to i. Therefore, we only need to prove that $\mathbf{a}_i = \mathbf{a}_{T(i,\theta)}$, i.e., $\mathbf{A}_{ij} = \mathbf{A}_{T(i,\theta)T(j,\theta)}$. According to Eq. (3), we denote $\mathbf{A}_{ij}$ as $h(\|\mathbf{v}_i - \mathbf{v}_j\|^2, \|p_i - p_j\|^2)$. Now, we first prove that $\|p_{T(i,\theta)} - p_{T(j,\theta)}\|^2$ equals $\|p_i - p_j\|^2$:

$$\begin{aligned} &\|p_{T(i,\theta)} - p_{T(j,\theta)}\|^2 \\ &= \|(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta) \\ &- (u\cos\theta - q\sin\theta, u\sin\theta + q\cos\theta)\|^2 \\ &= (x-u)^2 + (y-q)^2 \\ &= \|p_i - p_j\|^2. \end{aligned} \tag{11}$$

We then prove that $\|\mathbf{v}_{T(i,\theta)} - \mathbf{v}_{T(j,\theta)}\|^2$ equals $\|\mathbf{v}_i - \mathbf{v}_j\|^2$. According to Eq. (8), we have the following:

$$\begin{aligned} &\|\mathbf{v}_{T(i,\theta)} - \mathbf{v}_{T(j,\theta)}\|^2 \\ &= \|f(I_{T(i,\theta)}; \omega^{\mathbf{v}}) - f(I_{T(j,\theta)}; \omega^{\mathbf{v}})\|^2 \\ &= \|f(I_i; \omega^{\mathbf{v}}) - f(I_j; \omega^{\mathbf{v}})\|^2 \\ &= \|\mathbf{v}_i - \mathbf{v}_j\|^2 \end{aligned} \tag{12}$$

where the third line is obtained by incorporating Eq. (8) into the second line. In this way, we can obtain $g(I_{T(i,\theta)}) = T(g(I_i), \theta)$, which is occasionally called *rotational equivariance* and is quite beneficial for many types of tasks, such as document classification and semantic image segmentation.

6.2 Rotational Invariance Property

For the general classification task, it is desirable to learn a representation that is invariant to any viewpoint, i.e., rotational invariance. To achieve this goal, we propose simply performing a widely used yet efficient operation, i.e., global average pooling, on the rotational equivariance features obtained above. Specifically, our goal is to prove that $\mathcal{S}(I) = \mathcal{S}(T(I, \theta))$, where the function $\mathcal{S}(\cdot)$ denotes the final feature representation of our NCN. By incorporating the global average pooling, we have the following:

$$\begin{aligned} \mathcal{S}(I) &= \frac{1}{N}\sum_{i=1}^{N} g(I_i) \\ \mathcal{S}(T(I,\theta)) &= \frac{1}{N}\sum_{i=1}^{N} g(I_{T(i,\theta)}) \\ &= \frac{1}{N}\sum_{i=1}^{N} T(g(I_i), \theta) \\ &= \frac{1}{N} T(\sum_{i=1}^{N} g(I_i), \theta) \end{aligned} \tag{13}$$

where the third line is obtained from the second line based on the rotational equivariance obtained above. Because $\sum_{i=1}^{N} g(I_i)$ denotes an image with a single pixel, $T(\sum_{i=1}^{N} g(I_i), \theta) = \sum_{i=1}^{N} g(I_i)$. Hence, $\mathcal{S}(I) = \mathcal{S}(T(I, \theta))$, which proves that the rotation cannot change the feature representation of our NCL.

6.3 More Ablation Study

To ablatively analyze data augmentation, we have augmented the training images with different magnitudes of rotations. The above results show that augmenting data may be helpful to CNN for handling the viewpoint change but it usually causes the lower recognition accuracy due to increasing the difficulty of pattern recognition. Without relying on the data augmentation, our NCN can well solve the novel view challenge.

Rotation Range		0	(-45, 45)	(-90, 90)	(-135, 135)	(-180, 180)
Normal View	CNN	92.4	90.2	86.4	84.1	81.3
	NCN	**93.6**	**91.0**	**85.4**	**84.1**	**82.4**
Normal View	CNN	40.5	41 .8	48.1	60.0	81.7
	NCN	**93.6**	**91.0**	**85.4**	**84.1**	**82.4**

Table 8: Comparisons of the expanded CIFAR-10 accuracy between CNN and NCN on both "normal" and "novel" viewpoints, where normal indicates that the model is evaluated on the original *test* set, while novel indicates that the model is evaluated on our simulated *test* set.