Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Residual Vector Capsule: Improving Capsule by Pose Attention

NING XIE[1], XIAOXIA WAN[2]
[1]School of Printing and Packaging, Wuhan University, Luojia Hill, Wuhan, 430072, China (e-mail: 395795046@qq.com)
[2]School of Printing and Packaging, Wuhan University, Luojia Hill, Wuhan, 430072, China (e-mail: wan@whu.edu.cn)

Corresponding author: XiaoXia Wan (e-mail: wan@whu.edu.cn).

ABSTRACT The convolutional neural network has significantly improved the accuracy of image recognition; however, it performs in a fragile manner when we apply viewpoint transformation or add noise to the image. Recent studies have proposed new neural networks named capsule. Capsule build the part-whole relationship in the entity through instantiation parameters, and they cluster instantiation parameters layer by layer through routing-by-agreement; therefore, capsule has stronger representational ability and robustness than convolutional neural networks. However, the routing-by-agreement of the capsule network is limited by the prior probability assumption, which performs in an unstable way in recognition accuracy and robustness. To remove the restriction of the prior probability assumption in the routing-by-agreement, we propose a new capsule named the residual vector capsule (RVC), which constructs the routing-by-agreement with self-attention. The experimental results show that compared with other capsule networks, RVC achieves competitive classification accuracy on MNIST, Fashion-MNIST, CIFAR-10 and SVHN, improves the viewpoint invariance of the model on SmallNorb, and significantly improves the robustness of the model against white box attacks on CIFAR-10 and SVHN.

INDEX TERMS Capsule network, self attention, adversarial robustness, viewpoint invariance

I. INTRODUCTION

Image recognition is an important topic in computer vision, and its purpose is to determine the object category for the entities of each image. Over the past few years, convolutional neural networks have significantly improved image recognition accuracy because they can approximate arbitrary distributions. However, there are many challenges for convolutional neural networks, such as spatial robustness and adversarial robustness. For example, for the same entity, the convolutional neural network performs well for the image from a certain observation angle, but a different observation angle will make it work incorrectly (Geirhos *et al.* [16], Logan Engstrom *et al.* [17]), which shows the fragile spatial robustness of the convolutional neural network. In another example, for an image, a convolutional neural network can achieve a very high recognition accuracy through training, but it will work incorrectly after applying carefully designed subtle perturbations to the image [18], Alexey Kurakin *et al.* [19], which shows the fragile adversarial robustness of the convolutional neural network.

To improve convolutional neural networks, a new neural network, the capsule, has been proposed in recent years, such as Sabour *et al.* [3] and Hinton *et al.* [4]. Different from convolutional neural networks, capsule represent entities by instantiation parameters. The instantiation parameters could be vectors or matrices that represent the entities' attributes, such as shape, size, direction and so on. Compared with convolutional neural networks, which represent entities by scalars, the instantiation parameters of the capsule network improve on the representational ability of neurons. By encoding the instantiation parameters layer by layer, the capsule network builds the part-whole relationship of the entity layer by layer and finally encodes the instantiation parameters of the whole entity. This method improves the convolutional neural network in two aspects. On the one hand, the instantiation parameters provide the viewpoint-invariant part-whole relationship [4] in the entities; therefore, for the entities from a new viewpoint, the model ir required to adjust only the instantiation parameters of the entities without additional training. On the other hand, according to relevant experiments, such as in Hahn *et al.* [6] and Hinton *et al.* [4], instantiation parameters also improve the adversarial robustness of the model.

The capsule represents the entities with the instantiation

parameters layer by layer. In each layer, it clusters the instantiation parameters with routing-by-agreement [3], [4] and encodes the probability according to the clustering result. However, routing-by-agreement has inherent defects; the most remarkable defect is that when the existing routing-by-agreement clusters the instantiation parameters, it must often make a prior probability assumption about the data distribution (such as the mixed Gaussian model in [4]). When the data distribution becomes complex, especially when the data is mixed with noise, the capsules may cluster the instantiation parameters together in strange ways [6]. In addition, capsule networks might not be able to learn the correct parse tree during training [6]. Hinton *et al.* [26] introduced that if the semantic object is imitated by the same capsule, then the capsule cannot predict the identity of the whole; otherwise, if they are shaped by different capsules, the similarities in their relationships to the whole cannot be captured.

Therefore, we believe that removing the prior probability assumptions in the routing-by-agreement can improve the performance of the capsule network on complex data. This approach will help the capsule network to construct the correct parse tree to construct the relationship between the information on the instantiation parameters and the probability.

In this paper, we propose a novel capsule called the residual vector capsule (RVC). The unitized instantiation parameter (we call it pose) and the length of the instantiation parameter (we call it probability) are explicitly separated from the instantiation parameter and explicitly construct the relationship of the pose and probability with self-attention [11]. Self attention can be regarded as a nonlocal operator, which constructs pose contexts and captures long-range dependencies [11] to evaluate the importance of each pose; it removes the prior probability assumptions of the routing-by-agreement. Self-attention of the pose is helpful for capsule networks to learn the correct parse tree, which improves the performance of the capsule network in real-world images.

In summary, the main contributions of this work are as follows:

1 We propose a new capsule, RVC, which establishes the relationship between the pose and probability by evaluating the long-term dependence relationship of the pose. RVC can be regarded as either a new self-attention or a structural transformation of the capsule routing.
2 RVC provides a new way to construct self-attention and constructs self-attention by the residual vector between poses in the form of a l_2 Gaussian distribution.
3 We verified the representational capability of RVC on MNIST, Fashion-MNIST, CIFAR-10/100 and SVHN. RVC has shown competitive classification accuracy for classification tasks on these datasets. Adversarial attack tests show that the adversarial robustness of RVC reaches state of the art. Multiviewpoint classification experiments in SmallNorb show that RVC has better viewpoint invariance. The reconstruction experiments under MNIST and Fashion-MNIST prove that RVC can also learn disentangled representations, which implies that the idea of RVC is correct.

II. BACKGROUND AND RELATED WORK

Different from the convolutional neural network's great success in image tasks, current studies also point out that convolutional neural networks have some fragility. Convolutional neural networks easily predict incorrect samples after spatial transformation [16], [17]. Logan Engstrom *et al.* [17] investigates the vulnerability of neural networks to translation and rotation samples, which fully demonstrates that spatial robustness as an independent property needs additional research. Geirhos *et al.* [16] investigated the performance of the deep neural network on distorted images. When the model is trained directly on distorted images, it performs very well for exact distortion but extremely poorly for other types of distortion. [16], [17] illustrate the weakness of the convolutional neural network in space robustness, and we believe that the viewpoint invariance [3], [4], [6] of the capsule network is a good entry point for the study of space robustness.

Due to the good prospects of the capsule network in representational learning, there have been many relevant studies [4], [6], [9], [10] since [3] proposed the capsule network. Sabour *et al.* [3] proposed that the capsule network is a group of active neurons that use vectors to represent the instantiation parameters of the entity and propose an iterative routing method, dynamic routing. Hinton *et al.* [4] proposed a matrix capsule, which represents the position relationship between the entity and the observer by matrix, and proposed a routing method based on a Gaussian mixture model. Both [3] and [4] are the pioneering works of the capsule; they represent entities by the instantiation parameter and build a parse tree by iterative routing. Gu *et al.* [10] replace the fundamental routing part of [3] with multihead attention-based graph pooling operations and create explanations for individual classifications effectively. De Sousa Ribeiro *et al.* [8] propose a new capsule routing algorithm derived from variational Bayes for fitting a mixture of transforming Gaussians and show that it is possible to transform the capsule network into a capsule VAE. Ahmed *et al.* [7] replace the iterative dynamic routing mechanism with the efficient attention module enhanced by differentiable binary routing. Hahn *et al.* [6] propose self-routing where each capsule is routing independently by its subordinate routing network; it performs more robustly against white-box adversarial attacks and affine transformations.

In addition, adversarial robustness is an important evaluation of the capsule network [3], [4], [6]. Adversarial attacks can be divided into white box attacks and black box attacks. White box attacks are an adversarial sample generation method for the network structure. Since this paper is aimed at improvement of the network structure, we focus on FGSM [18] and BIM [19]. Ian J. Goodfellow *et al.* [18] propose an adversarial sample generation method in which the depth model can be fooled by adding small random perturbations to the original image along the gradient ascent direction. Alexey

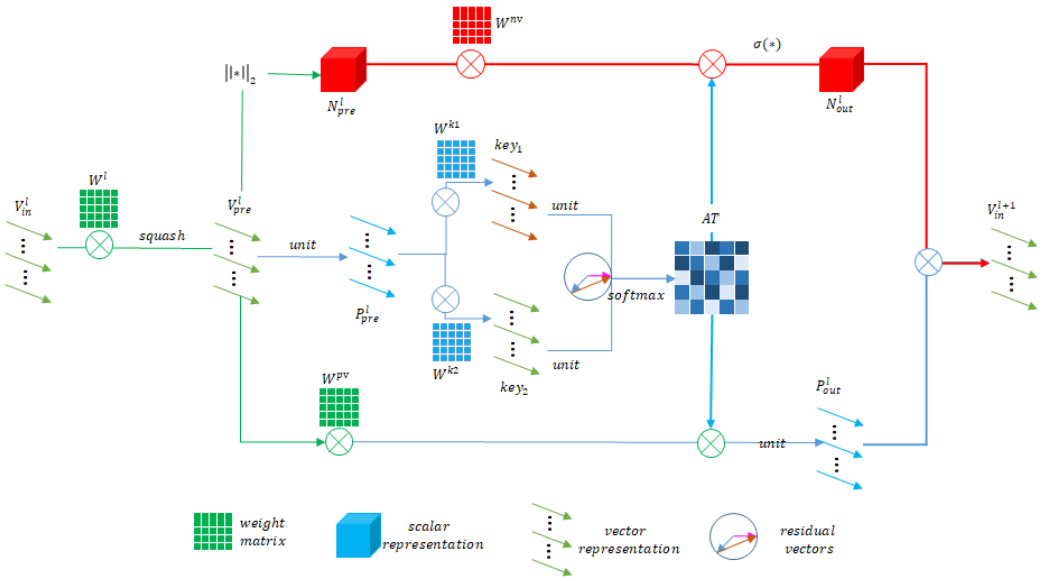


FIGURE 1. Overview of the residual vector capsule layer. The residual vector capsule layer is divided into two stages, prevoting and residual vector routing, without loss of generality, taking layer l as an example. The green path in the figure from V^l_{in} to N^l_{pre} and P^l_{pre} represents the prevoting stage; it aggregates the input into a prevote (V^l_{pre}) through the affine transformation matrix (W^l). The part-whole relationship in the entity is constructed layer by layer, and then, the prevote is separated into the probability (N^l_{pre}) and the pose (P^l_{pre}). The path from N^l_{pre} and P^l_{pre} to Vin^{l+1} represents the residual vector routing stage, and the residual vector attention (AT) is constructed in a unitized pose. Residual vector attention provides a global query mechanism for the pose and probability.

Kurakin *et al.* [19] improves [18], showing that a smaller ϵ after multiple iterations will generate stronger adversarial samples. Wu *et al.* [20] propose a novel mechanism to alleviate the overfitting issue and provide experimental results on a large dataset. As seen from the diagonal of Table 1 in [20], BIM remains one of the strongest adversarial attack methods.

Vaswani *et al.* [12] propose a new simple neural network module, the transformer, which is based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Xiaolong Wang *et al.* [11] present nonlocal operations as a generic family of building blocks for capturing long-range dependencies. Attention based on this approach is often called spatial self-attention. Zhang *et al.* [13] investigate the influence of global context information in the image segmentation task and propose the context coding module. Fu *et al.* [14] combined [13] with spatial attention [11] in the form of a dual attention network to complete semantic segmentation. Chi *et al.* [15] improves spatial attention different from [11]; it pools the length and width, which provides more efficient self-attention.

All attention mechanisms improve the model performance by building a context for data representation. In the form of $y = softmax(W^K X(W^Q X)^T)$, it can be regarded as multiple logistic regression, which is an estimation method without prior probability assumptions; therefore, we consider using self-attention to build the routing-by-agreement in the capsule network.

III. METHOD

The capsule networks represent the entities by instantiation parameters and cluster the instantiation parameters by routing-by-agreement. However, the prior probability assumption in the routing-by-agreement limits the performance of the capsule networks on complex data. We propose RVC (residual vector capsule), which represents the entities by the vectors and eliminates the restriction of the prior probability assumptions by constructing the self-attention of the poses. The design sketch is shown in Fig. 1. It consists of two stages: prevoting and residual vector attention. The prevoting aggregates and activates the lower level capsules to obtain the higher level capsules, and RVC establishes the part-whole relationship of the entities through the prevoting layer by layer. After prevoting, the residual vector attention establishes the self-attention of the pose in the higher-level capsule without a prior probability assumption and proves the global query mechanism for the probability and the pose.

The pre-voting builds the end-to-end architecture of the model by simulating the part-whole relationship in the entity, which provides a basic guarantee for the accuracy of the image recognition – the end-to-end architecture is easy to deepen, and the deeper the model is, the higher the accuracy is. Residual vector attention eliminates prior constraints in the routing-by-agreement, which makes it easier to construct parse trees, thus enhancing the model's adversarial robustness and viewpoint invariance.

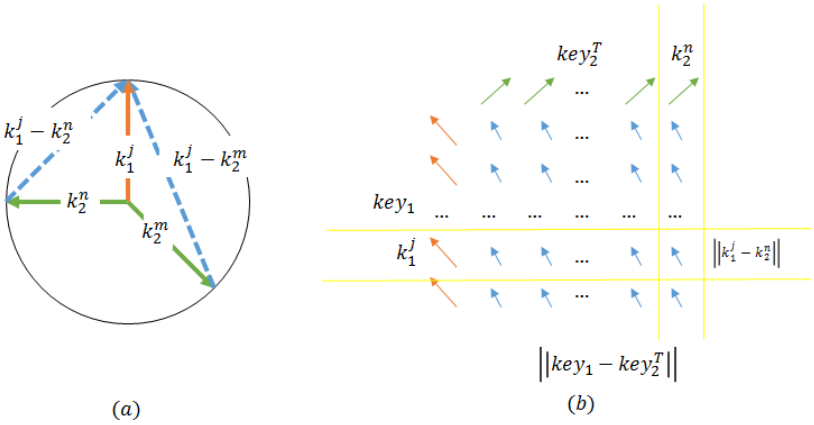


FIGURE 2. Key construction in residual vector routing. As shown in Fig. (a), the unitized key eliminates the influence of the vector length. During the training process, all of the vectors in the key rotate on the unitized circle. Here, $k_i^j \in key_i, i = 1, 2$ denotes the jth pose in key_i. In key_1 and key_2, the closer the orientation in the unitized circle is, the smaller the length of the residual vector will be. Fig. b denotes the construction of the residual vector attention, in which key_2^T denotes transposing key_2 on the channel dimension, and the length of the residual vector in $key_1 - key_2^T$ builds the pose context according to the orientation of the pose, which measures the consistency of the orientations of the poses. The closer the pose is, the more likely it is to indicate the existence of the entity.

A. PRE-VOTING

The purpose of prevoting is to construct the part-whole relationship of the entity; the instantiation parameters of the parts in the entity are aggregated layer by layer through affine transformation, and the model finally outputs the global instantiation parameters of the entity. Let $v_{(i)}^l \in V_{(in)}^l$ denote the ith capsule of the input in the lth layer, which denotes the local instantiation parameters of the entity. These instantiation parameters are viewpoint invariant in a single observation of the entity, and approximately constant, which means that the effect of the learnable parameter on the part of the entity is similar to the effect on the whole entity; thus, we assign a transformation matrix $W_{ij}^l \in R^{D_l \times D_{l+1}}$ to each v_i^l, where D_l denotes the vector dimension of $v_{(i)}^l$. We apply an affine transformation to $v_{(i)}^l$ by W_{ij}^l to simulate the part-whole viewpoint invariance and convert $v_{(i)}^l$ to the jth capsule in the $l+1$ layer:

$$v_{(j)}^l = \sigma(W_{ij}^l * v_{(i)}^l) \; where \; \sigma(v) = \frac{\|v\|^2}{\|v\|^2 + 1} \frac{v}{\|v\|} \quad (1)$$

, where $v_{(j)}^l$ denotes the jth capsule in V_{pre}^l, σ is the square [3], and $\|v\|$ denotes the length of the vector. (1) denotes the voting process from V_{in}^l to V_{pre}^l, $W_i j^l$ enables the network to learn the local-whole viewpoint invariance when aggregating votes; it applies an affine transformation layer by layer to the lower capsule to aggregate the vote and obtains the estimate of the higher level vote, and then, the squash function activates the higher capsules, squeezing the length of the vectors in each capsule to $(0, 1)$. The model builds the part-whole relationship layer by layer in this way.

In addition, we believe that prevoting can be regarded as two forms of representation – pose and probability. In the squash function, the pose is the unitized vector $\frac{*}{\|*\|_2}$; this vector represents the view point, size, direction and other pose information of the entity. The probability is the coefficient $\frac{\|*\|_2^2}{\|*\|_2^2+1}$, which represents the existence of an entity. For each pre-vote $v_{(j)}^l$, the separation of the entity probability from the pre-vote is simple:

$$n_{(j)}^l = \|v_{(j)}^l\|_2 \quad (2)$$

$$p_{(j)}^l = v_{(j)}^l / n_{(j)}^l \quad (3)$$

, where $n_{(j)}^l$ denotes the norm of the vector. It corresponds to $\frac{\|*\|_2^2}{\|*\|_2^2+1}$ in the squash function. $p_{(j)}^l$ denotes the aspect of the vector, which corresponds to $\frac{*}{\|*\|_2}$ in the squash function.

B. RESIDUAL VECTOR ROUTING

After the prevoting stage, the residual vector capsule enters the routing stage – residual vector routing. For higher-level capsules, residual vector routing provides a global query mechanism for the probability and pose in the form of self-attention. There are two advantages to using self-attention [12] as a routing. On the one hand, it removes the restriction of the prior probability assumption in the routing-by-agreement; on the other hand, the importance of all of the poses in the context is not the same, and the residual vector routing constructs the weight of each pose to measure its importance.

To construct the global query mechanism of the pose and probability, we must establish a relationship between the pose and probability. There is an observation: the closer the pose's orientation is, the more likely it is to indicate the existence of an entity, which comes from the reconstruction experiment in [3], [10], [22]. These reconstruction experiment settings are the same; the investigators applied small perturbations to a dimension in the vector representation (10 perturbations in [-0.25,0.25] by the interval of 0.05) and observed the differences in the generated results. According to the reconstruction experiment results in [3], [10], [22], the same capsule represents entities generated under subtle

perturbations that are semantically consistent. There is only a morphological difference – the smaller the disturbance is, the smaller the morphological difference, which shows that the capsules have learned the disentangled representation. Furthermore, we believe that under the decoupled representation of the capsule, on the one hand, the influence of a subtle perturbation on the vector length does not change the distribution learned by the model, and thus, a semantically consistent representation can be obtained. On the other hand, the influence of subtle perturbations on the directions of the vectors changes the morphology of semantics, and vectors with more similar directions suggest entities with more similar poses in the same semantics. Thus, it can be assumed that vectors that are closer in orientation are more likely to imply the existence of the same entity; in other words, in the context of pose, a closer pose is more likely to imply a higher probability.

Residual vector routing constructs self-attention in the form of $softmax(-\|W^Q X - W^K X^T\|_2^2)/\sqrt{(D/H)}$, which is similar to [27]'s formula 10, where $W^Q X, W^K X^T$ is called "key". Next, we first describe the construction of keys in vector residual routing, followed by the construction of attention.

Construction of key In Fig. 1, the construction of the key is represented as the blue paths from P^l_{pre} to key_1 and key_2. According to our observation of experiments [3], [10], [22], poses with similar directions suggest the existence of entities, and thus, the key point of the construction of the residual vector routing is to evaluate its distribution in terms of the pose direction. We construct the unitized key to eliminate the influence of the length of the vector. The length of the unitized vector is 1, and thus, the difference between the keys is in the orientation, but there is no difference in length. To measure the difference in orientation of the poses, for the pose $P^l_{pre} \in R^{D_l \times C_l}$, we define 2 learnable parameters $W^{k1}, W^{k2} \in R^{C_l \times C_l}$ to construct the unitized key:

$$key_i = \frac{W^{ki} * P^l_{pre}}{\|W^{ki} * P^l_{pre}\|_2}, i = 1, 2 \tag{4}$$

, where * denotes the matrix multiplication on the channel dimension, $\|W^{ki} * P^l_{pre}\|_2$ denotes calculating the length of every vector in $W^{ki} * P^l_{pre}$, to make every vector in $key_i \in R^{D_l C_l}$ be a unitized vector, and they are distributed on a unitized circle. Note that the length of the residual vector decreases monotonically with respect to the direction of the pose – when the two poses are in exactly the same direction, the length of the residual vector is 0; as the difference in the direction between the poses increases, the length of the residual vector monotonically increases; when the two vectors are in opposite directions, the length of the residual vector is 2. We then use the unitized keys to build self-attention.

Construction of attention In Fig. 1, the construction of attention is represented as a blue path between key_1 and key_2 to AT. The closer the orientation is, the more likely the pose is to indicate the existence of the entity; thus, the length of the residual vector presents a monotonically decreasing relationship with the orientation of the pose, as shown in Fig. 2b. The residual vector routing uses the length of the residual vector to measure the importance of each attitude in the context of the pose:

$$AT = softmax(-\frac{\|key_1 - key_2^T\|^2 W_\Phi}{\tau}) \tag{5}$$

, where $\|key_1 - key_2^T\|^2 \in R^{C_l \times C_l}$. As shown in Fig. 2b, it encodes the residual vectors' score field. For any two poses in key_1, key_2^T, if their orientation is close, the score that they made will be approximately 0. $W_\Phi \in R^{C_L C_L}$ is a learnable radius coefficient, and it extends the bounds of the unitized vectors in the score domain to $(-\infty, \infty)$. τ is a hyperparameter temperature to address gradient diffusion [12]. Softmax is applied to the column of the scoring domain to convert the scoring domain into a distribution. The closer the score is to 0 in the scoring domain, the higher the probability is in the distribution.

It should be emphasized that the essence of the residual vector attention AT is to construct the scoring domain according to the differences in the pose directions. It constructs the global query mechanism based on the length of the residual vector, which eliminates the restriction of the prior probability assumption in the capsule network and provides the global probability representation of the pose difference. In addition, the self-attention in the form of L2 is Lipschitz continuous [27], which is conducive to the convergence of the network. Therefore, the residual vector attention can be explicitly combined with the probability to make a decision:

$$n^l_j = \sigma(\sum_{i=1}^{C_l} AT_{i,j}(W^\phi * N^l_{pre})_j) \; where \; \sigma = \frac{x^2}{x^2+1} \tag{6}$$

, and where n^l_j denotes the jth probability value of N^l_{out} in Fig. 1, while $W^\phi * N^l_{pre}$ means feeding N^l_{pre} into a fully connected layer with weights W^ϕ. It should be emphasized that the probability representation in RVC combines the distribution on the AT column, which makes RVC incorporate the pose information context in the probability representation and remove the prior probability constraint. $\sigma(x) = x^2/(x^2+1)$ is a nonlinear activation function, which is consistent with squash [3].

In addition, an important feature of the capsule network is that it can obtain a series of instantiation parameters, such as the shape and appearance of the entity, and we also design the pose representation for residual vector routing. Note that even though the poses are all unitized vectors, they are not equally important to each other in the capsule, and thus, we use a prevote as the basis for the pose evaluation. The prevote can be regarded as the pose weighted according to probability, which implies the importance of each pose by probability evaluation. We use it in combination with AT to calculate the pose:

$$p^l_j = unit(\sum_{i=1}^{C_l}(AT_{i,j}(W^p * V^l_{pre})_j)) \tag{7}$$

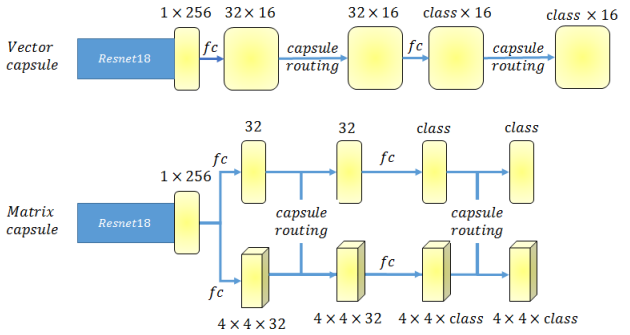


FIGURE 3. Decision layer consistency guarantee. As described by the backbone consistency guarantee, for any capsule, we stack two layers of the capsule layer after the global pooling layer of Resnet20. As stated in the consistency guarantee of the decision layer, the voting form of the matrix capsule is 4×4 dimensions, the voting form of the vector capsule is 16 dimensions, and the network's width is $256 \rightarrow 32 \rightarrow classes$.

, where $unit(X) = \{X_i' | X_i' = \frac{X_i}{\|X_i\|}\}$ denotes applying the unit to every vector in X_i,p_j^l denotes the jth pose in P_{out}^l in Fig. 1, and $W^p * V_{pre}^l$ means feeding V_{pre}^l into a fully connected layer with the weight $W^p \in R^{C_l \times C_l}$. Therefore, p_j^l can be regarded as a normalized evaluation of the entity instantiation parameter. Following squash [3], the output of the residual vector routing is

$$v_j^{l+1} = n_j^l p_j^l \tag{8}$$

; v_j^{L+1} is the jth vote of V_{in}^{L+1} in Fig. 1. Combining (7) and (8), it can be seen that the purpose of Equation (9) is to obtain the importance of each pose according to probability weighting.

IV. EXPERIMENTS

The difference between the capsule and convolutional neural network is that it shows more features, such as adversarial robustness, viewpoint invariance, and disentangled representation. In this paper, we provide a comprehensive evaluation of RVC, which includes the following:

1. Classified performance. We compare RVC with baseline capsule networks by the image classification task.

2. Adversarial robustness. We compare RVC with the previous capsule networks by adversarial-attack experiments.

3. Viewpoint invariance. We verify the viewpoint invariance of RVC on SmallNorb.

4. Disentangled representation. We show that RVC also learns disentangled representations by image reconstruction experiments.

In this section, we first give the experimental configuration. Subsection 4.1 introduces the consistency guarantee of the comparative experiment, subsection 4.2 introduces the experimental results of image classification of each model, subsection 4.3 introduces the robustness of each model under the contrast sample, and subsection 4.4 introduces the viewpoint invariance of each model. Finally, subsection 4.5 introduces the reconstruction experiment of RVC.

A. EXPERIMENT CONSISTENCY GUARANTEE

We run the experiments on a GTX1060 6G. For the fairness of the experiments, we must conduct some guarantees, including dataset consistency guarantee, backbone consistency guarantee, decision layer consistency guarantee, training process consistency guarantee, and adversarial test consistency guarantee.

Dataset consistency guarantee Most of the capsule networks evaluate the model performance with small data sets [3], [4], [6], [10]. We choose MNIST, Fashion-MNIST, CIFAR10/100, SVHN and SmallNORB as the test benchmarks. MNIST and Fashion-MNIST are grayscale image datasets with 50000 training samples and 10000 test samples. We resize the original image to 32×32 for training, and they are used for the classification experiment and reconstruction experiment of the disentangled representation. CIFAR10/100 is a tiny image dataset with 10 classes, which contains 50000 training samples and 10000 test samples in total. We used random crops, random horizontal flips, and normalization as data augmentation policies. The SVHN dataset was obtained from a large number of street view images using a combination of automated algorithms and the Amazon Mechanical Turk (AMT) framework, which was used to localize and transcribe the single digits. It contains 73257 training samples and 26032 test samples, and we use random crop and normalization as the data augmentation policies. CIFAR10/100 and SVHN were used for classification experiments and adversarial attack experiments. Following [4], SmallNORB was used to evaluate the viewpoint invariance, and we used the same data augmentation policy as in [4]. The familiar experiments are performed by training and testing on all viewpoints, while the novel experiments are performed by holding out the most unique azimuths (from 6 to 28) and elevations (from 3 to 8).

Backbone consistency guarantee Resnet20 [23] is used in all of the comparison methods as the backbone; it has been repeatedly validated in small data sets [3], [4], [6], [7], on which it performs well, and we set the channels of the first convolutional layer to 32. In addition, there is an obvious identification between the convolution layer and the decision layer in ResNet20 – the global average pooling layer. For all models, we connected the capsule module as the decision layer after the global average pooling layer.

TABLE 1. The classification accuracy of RVC on MNIST, Fashion-MNIST, CIFAR-10 and SVHN.

	MNIST	Fashion-MNIST	CIFAR-10	CIFAR-100	SVHN
Resnet-18	99.02%	92.5%	92.06%	69.15%	95.58%
Dynamic Routing [3]	99.15%	92.8%	91.56%	52.11%	95.91%
EM Routing [4]	97.25%	91.6%	90.46%	39.05%	93.06%
Self-Routing [6]	99.24%	94.01%	**92.03%**	69.23%	96.07%
IDPA-Caps [5]	99.15%	93.1%	92.01%	69.52%	96.70%
RVC	**99.32%**	**94.13%**	91.70%	**70.69%**	**97.03%**

Decision layer consistency guarantee To guarantee the fairness of the comparison between capsule models, we limited the capsule to the same size of computing space in all comparative experiments. Considering that the output of the global average pooling layer is a one-dimensional vector, we access all capsule modules in the form of full connections; in addition, vector capsules are 16 dimensions, and matrix capsules are 4×4 dimensions. For all of the data sets, we evaluate the performance using a two-layer capsule, as described in Fig. 3.

Loss function and training process consistency guarantee For all of the comparison models [3], [4], [6], [10], we use the same loss function as the original paper to train, and for the RVC, we use margin loss training:

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(\|v_k\| - m^-, 0)^2 \tag{9}$$

, where $T_k = 1$ if the object of the kth class is present. As in [3], the hyperparameters are often empirically set as $m^+ = 0.9, m^- = 0.1$ and $\lambda = 0.5$.

In the training, we found that the convergence rates of the matrix model and vector model were not consistent, and thus, we compare all models by training for 90 epochs. Specifically, for all models, we use SGD in training, set the weight decay to $5x10^{-4}$, set the initial step size to 0.1, and every 30 epochs the decay is set to 1/10 of the current step size.

In particular, with EM Routing [4], to make the spread loss more stable, we trained an overfitting convolutional neural network on the data set first and then finetuned the weight of the backbone to the model. To be fair in comparison, we applied this trick to all models.

Adversarial test consistency guarantee We use FGSM [18] and BIM [19] to test the adversarial robustness on SVHN and CIFAR10 for all models. In relevant works, nondirected attacks are often stronger than directed attacks (for example, Fig. 3 of [6]), and we chose nondirected attacks to complete the adversarial robustness test. For FGSM, we set $\epsilon = 0.1$ in eq. 1, and for BIM, we set the iterations to 10 times (namely, step 0.01 FGSM performs 10 times).

B. CLASSIFICATION PERFORMANCE

To measure the representational ability of each capsule network, we designed an image classification experiment according to the experimental guarantee as in subsection 4.1. Considering that we have only one GTX1060 and that most capsule network studies [3], [4], [10] evaluate the model on small data sets, we select MNIST, F-MNIST, CIFAR10/100 and SVHN as the criteria to evaluate the classification ability of the model. In addition, [3] and [4] are pioneering studies on capsule networks. [6] builds the neural network without routing-by-agreement; as a result, it does not need a prior probability assumption. [5] is the recent research result of the capsule network, and thus, we choose these five as the comparison benchmark.

As shown in Table 1, relatively speaking, most of the capsule networks will reduce the representational ability of the backbone, and the performance of RVC is the same as that of the backbone. On the grayscale image dataset (MNIST, fashion-MNIST), the distribution of data patterns is relatively simple, and all of the models fit quite well. In the complex data sets (CIFAR-10, SVHN), RVC shows a competitive performance on the whole, which performs the same as the backbone. Compared with other data sets, the data distribution of CIFAR100 is much more complex, and the performance of some capsule networks drops sharply, while RVC still shows a classification ability similar to the backbone.

C. ADVERSARIAL ROBUSTNESS

In this subsection, we use the classification of models in subsection 4.2, and we compare our model with other routing algorithms in terms of adversarial robustness. Adversarial attacks can be divided into black box attacks and white box attacks. The black box attack has no business with the network structure. Considering that the residual vector capsule is aimed at the improvement of the network structure, we select the white box attack as the test method and use FGSM [18] and BIM [19] to generate adversarial samples. Their hyperparameter settings are the same as [10], and the same settings are used to attack all models. Instead of choosing a single perturbation threshold, we use different thresholds to show a curve, i.e., in the range $[0.01, 0.05]$ with the interval of 0.01.

Fig. 4 shows the performance of the convolutional neural network (backbone + avg) and 4 capsules (DR, EM, SR and residual vector capsule) under two adversarial attack methods. All of the capsule modules can improve the robustness of the convolutional neural network, and the residual vector capsule has the greatest improvement. Especially for CIFAR10, when BIM takes the 0.5 threshold value, more than half of the convolutional neural network's predictions

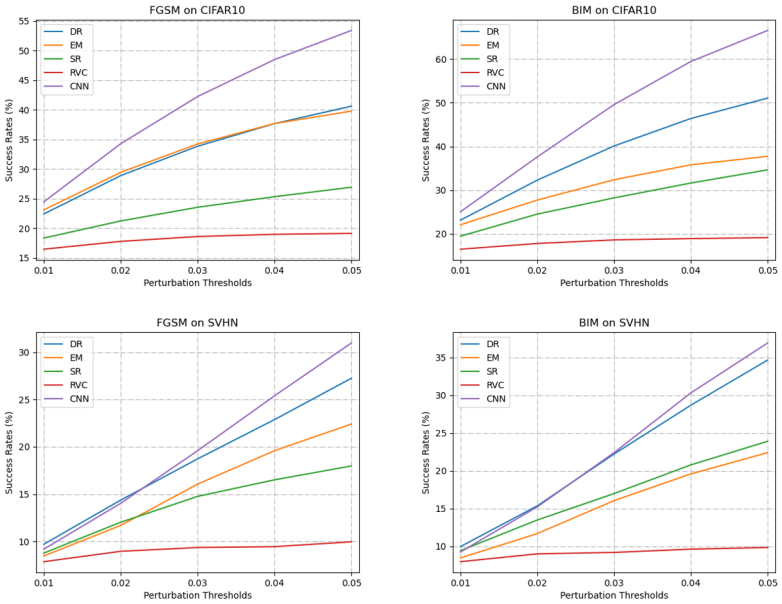


FIGURE 4. The attack success rate of FGSM and BIM against each model; the lower the attack success rate is, the stronger the adversarial robustness is. The attack methods attack our models (RVC) with a lower success rate.

TABLE 2. Error rates of similar and new viewpoints on SmallNorb. Novel denotes the error rate of a new viewpoint, familiar denotes the error rate of similar viewpoints.

	Azimuth		Elevation	
	Familiar(%)	Novel(%)	Familiar(%)	Novel(%)
Resnet-18	8.42±0.48	22.43±1.32	7.82±0.63	18.97±0.92
Dynamic Routing [3]	8.28±0.50	19.33±1.02	7.57±0.46	17.18±0.88
EM Routing [4]	7.25±0.68	14.11±0.98	6.39±0.81	12.73±0.80
Self-Routing [6]	7.85±0.61	18.47±0.97	6.89±0.72	16.62±1.29
RVC	**7.92±0.71**	**9.63±0.21**	7.22±0.81	**9.52±0.93**

are misled, while RVC remains at a low error rate while under attack. Furthermore, not only is the result improved, but the error rate of RVC at each threshold grows significantly slower than that of all of the other capsule networks. We believe that this finding occurs because the residual vector capsules incorporate pose difference information in the probability, and the pose difference information provides richer semantic object information, which makes the model more robust under all adversarial attacks.

D. VIEWPOINT INVARIANCE

Another difference between the capsule networks and the convolutional neural networks is the viewpoint invariance. For an object, its relationship to the observer can be represented by instantiation parameters without additional data sampling or neuron activation [9] in the capsule networks. To measure the ability of the RVC to represent different viewpoints, we designed a viewpoint invariance experiment on the SmallNorb dataset, using the same architectures as subsection 4.2 to complete the experiment. We used two indices to evaluate the experimental results: the accuracy of similar viewpoints and the accuracy of new viewpoints. The accuracy of a similar viewpoint was the performance of the model when it was trained and tested on all viewpoints in SmallNorb, while the accuracy of the new viewpoint was the performance of the model when it was trained and tested according to the setting of SmallNorb in subsection 4.1's dataset consistency guarantee.

As shown in Table 2, compared with convolutional neural networks, all of the capsule networks have higher accuracy for the new viewpoint, and RVC has achieved better performance. Compared with other capsule networks, the performance of RVC on the horizontal view point (azimuth) is 5%-8%, and the performance on the vertical viewpoint (elevation) is 3%-8%, which indicates that the new viewpoint invariance of the capsule networks can be improved by adding pose difference information into the probability.

E. DISENTANGLED REPRESENTATION

Disentangled representation is another feature of capsule networks. After the RVC encoder learns the disentangled representation, we apply a reconstructed network to the RVC encoder. When a slight perturbation is applied to the components of the disentangled representation, the reconstructed

image will change. The idea of RVC came from an observation in the reconstruction experiment in [3], [10], [22]: the vectors that are closer in direction are more likely to indicate the existence of an entity. Therefore, we also apply a reconstruction experiment to RVC, and the settings of the reconstruction experiment were the same as in [3], [10], [22]. We apply perturbations to a dimension of the vector on the interval [-0.25,0.25] and take 0.05 as the perturbation interval, to observe its reconstruction representation.

As seen from Tables 3 and 4, the residual vector capsule can learn the disentangled representation, and the perturbations on various dimensions can affect the entity's shadow, thickness, style, and more, but do not change the entity's semantics. This finding experimentally confirms our idea that vectors that are closer in direction are more likely to indicate the existence of an entity.

TABLE 3. MNIST Dataset. The disentangled representation of RVC on MNIST

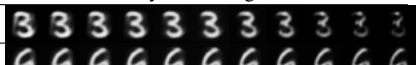
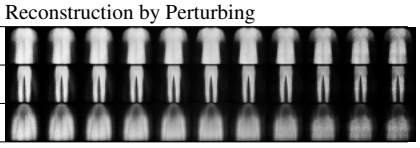
class	Property	Reconstruction by Perturbing
3	shadow	
6	Thickness	
7	Style	

TABLE 4. Fashion-MNIST Dataset, the disentangled representation of RVC on Fashion-MNIST

class	Property	Reconstruction by Perturbing
Dress	Thickness	
Trouser	Style	
Bag	Height	

V. CONCLUSIONS

In this paper, we propose a new capsule network, the residual vector capsule. By separating the pose and probability during prevoting, we construct a routing mechanism based on self-attention, which eliminates the influence of prior probability assumptions in routing-by-agreement. The experiment proves that compared with the current work, the residual vector capsule achieves competitive classification accuracy, slightly improves the viewpoint invariance, and significantly improves the adversarial robustness. Moreover, we also believe that this explicit separation method of probability and pose is an idea that is worthwhile to study.

REFERENCES

[1] David, G., "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision,* vol. 147, Jan. 2004, pp. 91-110.

[2] Bay, Herbert and Tuytelaars, Tinne and Van Gool, Luc, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV 2006,* pp. 404–417.

[3] Sabour, Sara and Frosst, Nicholas and Hinton, Geoffrey E, "Dynamic Routing Between Capsules" in *Advances in Neural Information Processing Systems 30,* 2017, pp. 3856–3866.

[4] Geoffrey E Hinton and Sara Sabour and Nicholas Frosst, "Matrix capsules with EM routing" in *International Conference on Learning Representations,* 2018.

[5] Yao-Hung Hubert Tsai and Nitish Srivastava and Hanlin Goh and Ruslan Salakhutdinov, "Capsules with Inverted Dot-Product Attention Routing," in arXiv 2002.04764, Available: https://arxiv.org/abs/2002.04764.

[6] Hahn, Taeyoung and Pyeon, Myeongjang and Kim, Gunhee, "Self-Routing Capsule Networks," in *Advances in Neural Information Processing Systems 32,* 2019, pp. 7658–7667.

[7] Ahmed, Karim and Torresani, Lorenzo, "STAR-Caps: Capsule Networks with Straight-Through Attentive Routing," in *Advances in Neural Information Processing Systems 32,* 2019.

[8] De Sousa Ribeiro, F., Leontidis, G., Kollias, S., "Capsule Routing via Variational Bayes," in *Proceedings of the AAAI Conference on Artificial Intelligence 34th*. pp. 3749–3756

[9] Kosiorek, Adam and Sabour, Sara and Teh, Yee Whye and Hinton, Geoffrey E, "Stacked Capsule Autoencoders," in *Advances in Neural Information Processing Systems 32,* 2019, pp. 15512–15522.

[10] Gu, Jindong, "Interpretable Graph Capsule Networks for Object Recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence 34th*. 2020.

[11] Xiaolong Wang and Ross Girshick and Abhinav Gupta and Kaiming He, "Non-local Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[12] Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Lukasz and Polosukhin, Illia, "Attention is All you Need," in *Advances in Neural Information Processing Systems 30,* 2017.

[13] Zhang, Hang and Dana, Kristin and Shi, Jianping and Zhang, Zhongyue and Wang, Xiaogang and Tyagi, Ambrish and Agrawal, Amit, "Context Encoding for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June. 2018.

[14] Fu, Jun and Liu, Jing and Tian, Haijie and Li, Yong and Bao, Yongjun and Fang, Zhiwei and Lu, Hanqing, "Dual Attention Network for Scene Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June. 2019.

[15] Chi, Lu and Yuan, Zehuan and Mu, Yadong and Wang, Changhu, "Non-Local Neural Networks With Grouped Bilinear Attentional Transforms," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June. 2020.

[16] Geirhos, Robert and Temme, Carlos R. M. and Rauber, Jonas and Schütt, Heiko H. and Bethge, Matthias and Wichmann, Felix A., "Generalisation in humans and deep neural networks," in *Advances in Neural Information Processing Systems 31,* 2018.

[17] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, Aleksander Madry, "Exploring the Landscape of Spatial Robustness," in *International Conference on Machine Learning,36th,* 2019.

[18] Ian J. Goodfellow and Jonathon Shlens and Christian Szegedy, "Explaining and Harnessing Adversarial Examples," in *International Conference on Learning Representations, 5th,* May 7-9, 2015.

[19] Alexey Kurakin and Ian J. Goodfellow and Samy Bengio, "Adversarial examples in the physical world," in *arXiv, 1607.02533,* 2016.

[20] Wu, Weibin and Su, Yuxin and Chen, Xixian and Zhao, Shenglin and King, Irwin and Lyu, Michael and Tai, Yu-Wing, "Boosting the Transferability of Adversarial Samples via Attention," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* June, 2020. pp. 1158-1167.

[21] Pang, Tianyu and Du, Chao and Zhu, Jun, "Max-Mahalanobis Linear Discriminant Analysis Networks," in *Proceedings of the 35th International Conference on Machine Learning,* 2018. pp. 4016–4025.

[22] Rajasegaran, Jathushan and Jayasundara, Vinoj and Jayasekara, Sandaru and Jayasekara, Hirunima and Seneviratne, Suranga and Rodrigo, Ranga, "DeepCaps: Going Deeper With Capsule Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June. 2019.

[23] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, "Identity Mappings in Deep Residual Networks," in *arXiv:1603.05027*. 2016.

[24] Radosavovic, Ilija and Kosaraju, Raj Prateek and Girshick, Ross and He, Kaiming and Dollar, Piotr, "Designing Network Design Spaces," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June. 2020.

[25] Mingxing Tan, Quoc Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*. 2019.

[26] Geoffrey Hinton, "How to represent part-whole hierarchies in a neural network," in *arXiv:2102.12627*. 2021.

[27] Hyunjik Kim and George Papamakarios and Andriy Mnih, "The Lipschitz Constant of Self-Attention," in *arXiv:2006.04710*. 2020.