Capsule Network is Not More Robust than Convolutional Network

(Supplementary Materials)

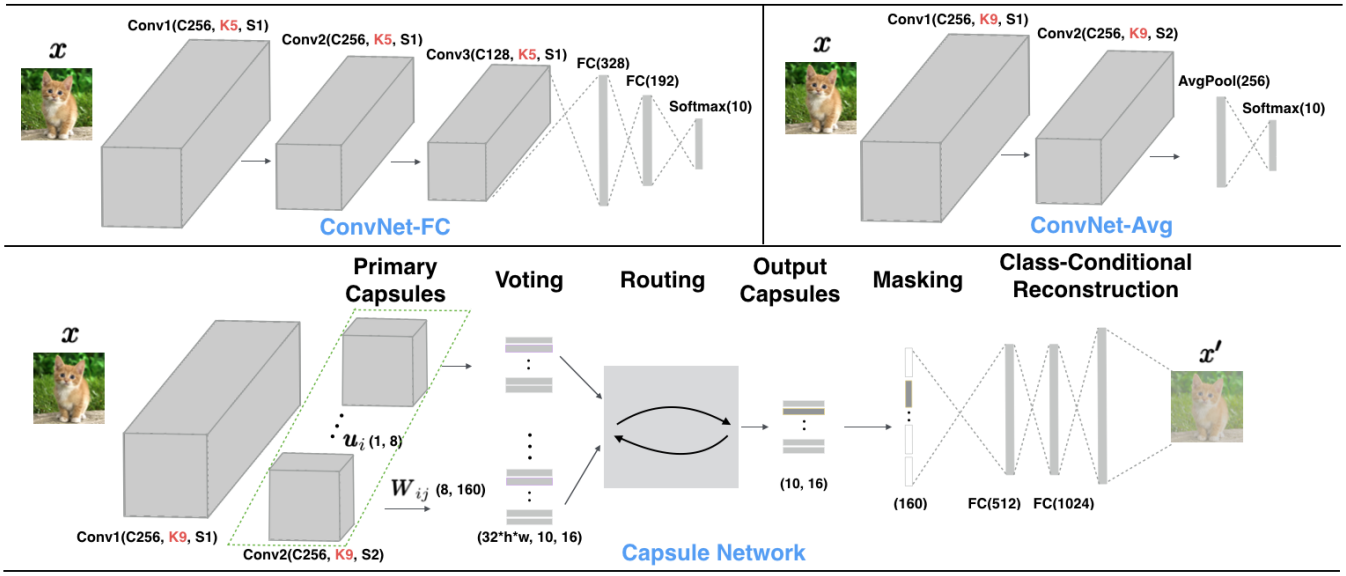


Figure 1: The overview of ConvNet and CapsNet architectures: The kernel sizes are marked with red color, and the capsule types (the groups) are marked with a dashed green box.

1. Supplement A: the effect of the number of capsule groups and the capsule sizes

In Section 3.1 of the paper, we study the effect of components of CapsNet on its affine transformation robustness. This supplement section shows the effect of architecture configurations on the transformation robustness, namely, the number of capsule dimensions (capsule size) and the number of capsule groups (marked with a green box in Fig. 1).

Factors	Num-groups	Caps-size	Routing	SharedM	Squash-fn	Reconstion	Loss	Test-MNIST	Test-AffNIST
Num-groups	1	8	NoR	✓	squash	✓	MarginLoss	99.26(± 0.27)	93.73(± 0.51)
	2	-	-	-	-	-	-	99.21(± 0.44)	92.92(± 0.62)
	8	-	-	-	-	-	-	99.45(± 0.38)	94.03(± 0.31)
	32	-	-	-	-	-	-	99.32(± 0.23)	93.55(± 0.39)
Caps-size	32	8	NoR	✓	squash	✓	MarginLoss	99.28(± 0.33)	93.55(± 0.61)
	16	16	-	-	-	-	-	99.24(± 0.31)	93.99(± 0.24)
	8	32	-	-	-	-	-	99.19(± 0.49)	94.01(± 0.37)

Table 1: The CapsNets with various architectures are trained only on MNIST dataset. The performance on MNIST training dataset, MNIST test dataset, and AffMNIST test dataset are reported, respectively (in percentage %). When CapsNet architecture is configured differently, the corresponding performance is reported. Given the variance, the number of capsule groups and the capsule size have no effect on the transformation robustness of CapsNet.

The performance on MNIST training data, MNIST test data, and AffNIST test data are reported in Tab. 1. The different architecture configurations with the same parameters make no difference in the generalization performance, given the variance. This study shows that the number of capsule dimensions and the number of capsule groups have no effect on the generalization ability of CapsNet to input affine transformations.

2. Supplement B: the effect of the kernel sizes on the ability to recognize overlapping digits

This study investigates the effect of the kernel sizes on the overlapping digits recognization ability of different models, such as ConvNet-FC and CapsNet. The performance of the models with different kernel sizes is reported in Tab. 2. To be noted that the model size of ConvNet-FC becomes smaller when large kernels are applied. The reason behind this is that the large kernels lead to smaller feature maps in case of no padding, which further leads to smaller units in the fully connected layer in ConvNet-FC. In CapsNet, the smaller feature maps lead to smaller transformation matrices.

Kernels	K(3, 3)		K(5, 5)		K(7, 7)		**K(9, 9)**		K(11, 11)	
Models	#Para.	A_{std}	#Para.	A_{std}	#Para.	A_{std}	#Para.	A_{std}	#Para.	A_{std}
CapsNet	13.0M	76.01	11.6M	78.92	11.1M	79.65	11.4M	80.22	12.5M	81.97
ConvNet-FC	38.7M	**85.23**	26.7M	**85.78**	18.5M	**85.77**	14.1M	**85.19**	13.5M	**85.34**

Table 2: The effect of the kernel sizes on the overlapping digits recognization ability of different models: The application of large kernels can improve the model's ability to recognize overlapping digits. ConvNet-FC outperforms CapsNet, even when the same model size is kept.

In the table, given a kernel size and the data size ($\times 10$), we report both the model size and the accuracy to classify overlapping digits of each model. ConvNet-FC outperforms CapsNet on this overlapping digits recognization task when the same kernel size is applied. Especially, when the kernel size 9×9 is applied, the model sizes of ConvNet-FC and CapsNet are similar, and ConvNet-FC outperforms CapsNet by 3.37%.

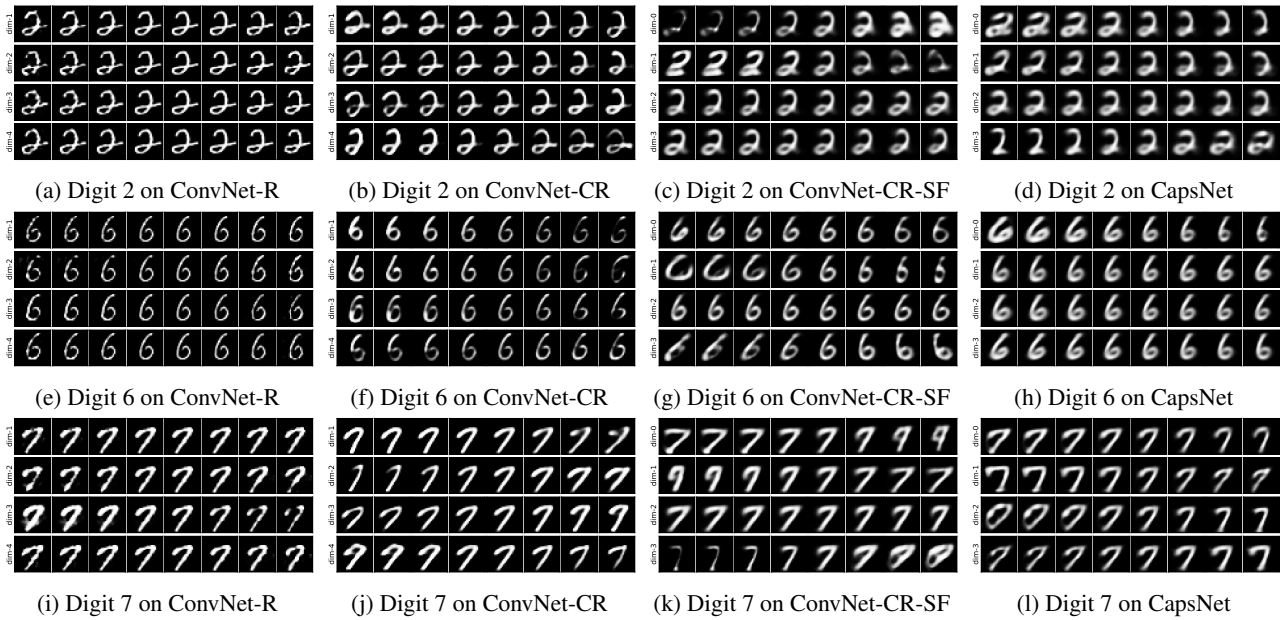
3. Supplement C: Semantic Representations

In CapsNets, when a single element of the vector representation is perturbated, the reconstructed images are also visually changed correspondingly. We conduct the same experiment on different models we build, namely, ConvNet-R, ConvNet-CR, and ConvNet-CR-SF as well as CapsNet. The more figures on MNIST dataset are shown in Fig. 2.

In addition, we also verify our claims on FMNIST dataset. Similarly, we visualize the reconstructed images under different perturbations in Fig. 3. We also report the semantic compactness score of the learned representations in Tab. 3. Given the perturbation range, we also reduce the perturbation interval to show more intermediate images in Fig. 4 and Fig. 5. All the visualizations, as well as the table, show consistent results with the ones on MNIST. Namely, both the class-conditional reconstruction mechanism and the squashing function can help ConvNet learn meaningful semantic representations.

Datasets	MNIST						FMNIST					
Factors	Rotation	Trans-X	Trans-Y	Scale	Shear-X	Shear-Y	Rotation	Trans-X	Trans-Y	Scale	Shear-X	Shear-Y
CNN-R	0.0003	0.0016	0.0009	0.0004	0.0003	0.0007	0.0005	0.0004	0.0006	0.0005	0.0009	0.0002
CNN-CR	0.0028	0.0038	0.0032	0.0052	0.0058	0.0022	0.0038	0.0019	0.0012	0.0016	0.0042	0.0031
CNN-CR-SF	**0.0325**	**0.2010**	**0.3192**	**0.0146**	**0.0476**	**0.0506**	**0.0062**	**0.0078**	**0.0233**	**0.0092**	**0.0074**	**0.0159**
CapsNet	0.0031	0.0107	0.0464	0.0026	0.0098	0.0021	0.0018	0.0017	0.0022	0.0013	0.0022	0.0018

Table 3: The representation compactness: The class-conditional reconstruction and the squashing function improve the compactness, while dynamic routing reduces it. This claim is true on both MNIST and FMNIST datasets.

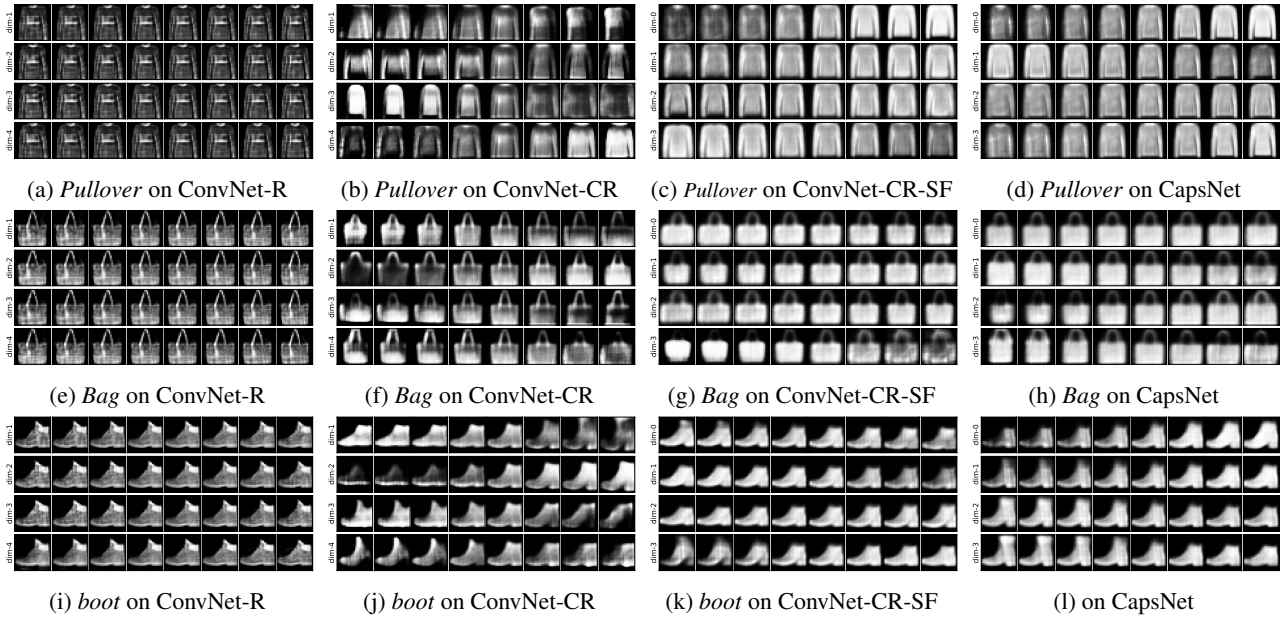


(a) Digit 2 on ConvNet-R (b) Digit 2 on ConvNet-CR (c) Digit 2 on ConvNet-CR-SF (d) Digit 2 on CapsNet

(e) Digit 6 on ConvNet-R (f) Digit 6 on ConvNet-CR (g) Digit 6 on ConvNet-CR-SF (h) Digit 6 on CapsNet

(i) Digit 7 on ConvNet-R (j) Digit 7 on ConvNet-CR (k) Digit 7 on ConvNet-CR-SF (l) Digit 7 on CapsNet

Figure 2: The reconstructed images on MNIST dataset are shown when a single unit of representation is perturbed. We show the images on some classes where images and classes are selected randomly. The reconstruction only helps when the class-conditional masking mechanism is applied. The squashing function improves the visual response further.



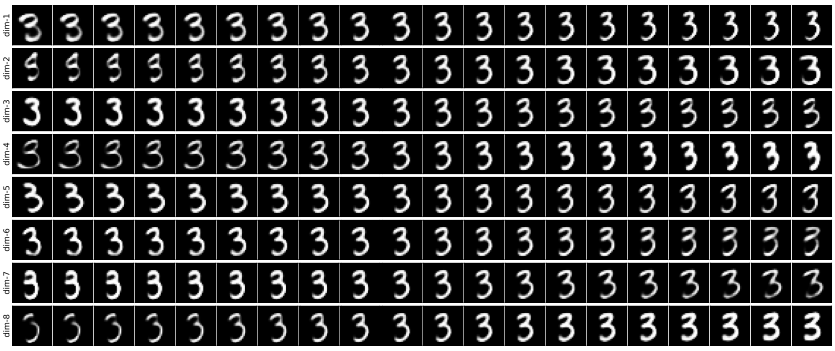
(a) *Pullover* on ConvNet-R (b) *Pullover* on ConvNet-CR (c) *Pullover* on ConvNet-CR-SF (d) *Pullover* on CapsNet

(e) *Bag* on ConvNet-R (f) *Bag* on ConvNet-CR (g) *Bag* on ConvNet-CR-SF (h) *Bag* on CapsNet

(i) *boot* on ConvNet-R (j) *boot* on ConvNet-CR (k) *boot* on ConvNet-CR-SF (l) on CapsNet

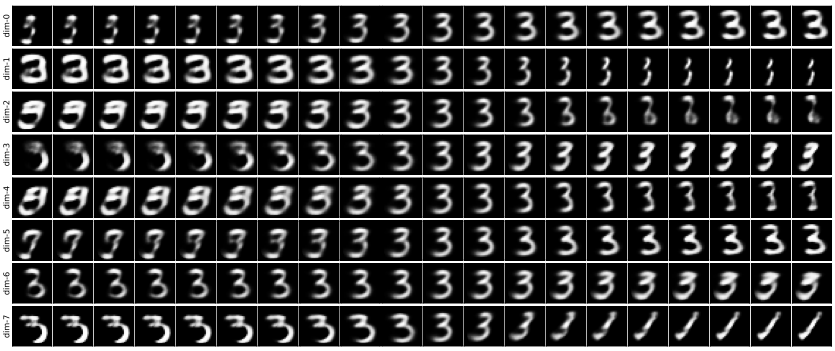
Figure 3: The reconstructed images on FMNIST dataset are shown when a single unit of representation is perturbed. We show the images on some classes where images and classes are selected randomly. They are *Pullover*, *Bag*, and *boot*. The observation is consitent with the one on MNIST.

(a) on ConvNet-R



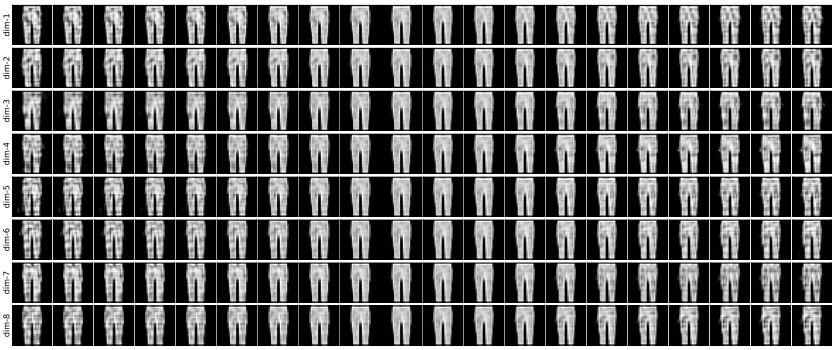
(b) on ConvNet-CR

(c) on ConvNet-CR-SF

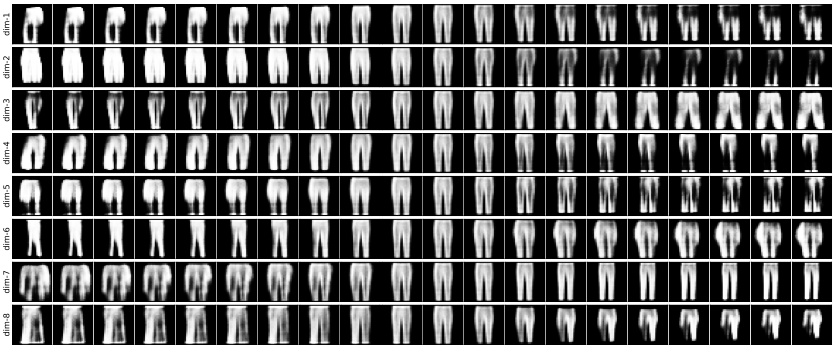


(d) on CapsNet

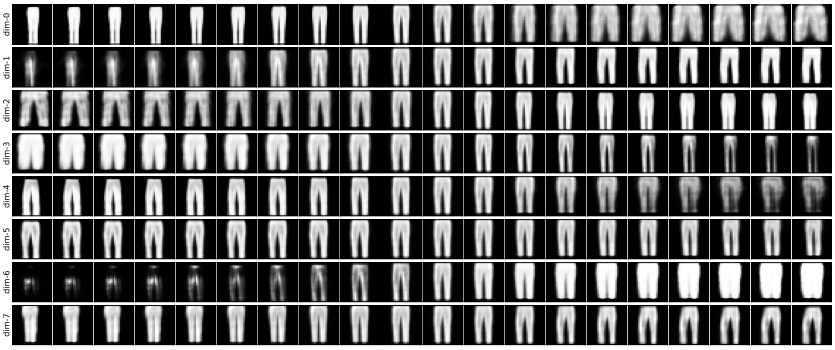
Figure 4: The reconstructed images on MNIST dataset are shown when a single unit of representation is perturbed. We reduce the perturbation interval to obtain more reconstructed images. The conclusion is the same. Namely, the reconstruction only helps when the class-conditional masking mechanism is applied, and the squashing function improves the visual response further.



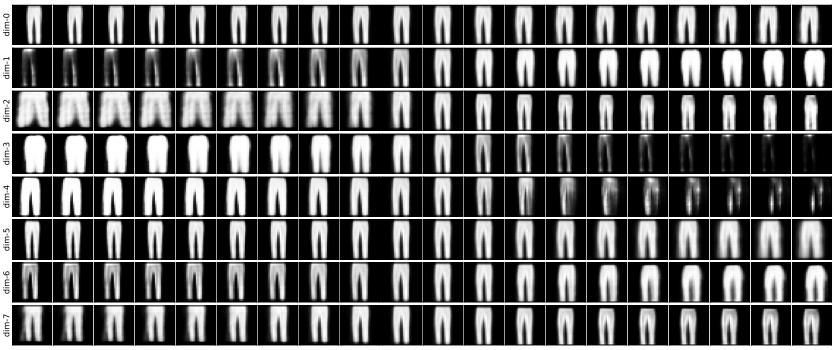
(a) on ConvNet-R



(b) on ConvNet-CR



(c) on ConvNet-CR-SF



(d) on CapsNet

Figure 5: The reconstructed images on FMNIST dataset are shown when a single unit of representation is perturbed. The reconstruction only helps when the class-conditional masking mechanism is applied. The squashing function improves the visual response further.