Are Capsule Networks Texture or Shape Biased?

Riccardo Renzulli
Department of Computer Science
University of Turin, Italy
riccardo.renzulli@unito.it

Dominik Vranay
Department of Cybernetics and Artificial Intelligence
Technical University of Košice, Slovakia
dominik.vranay@tuke.sk

Marco Grangetto
Department of Computer Science
University of Turin, Italy
marco.grangetto@unito.it

Abstract

Capsule networks (CapsNets) have been proposed as an alternative to traditional convolutional neural networks (CNNs), with the promise of better capturing part-whole relationships and spatial hierarchies. While CNNs are known to exhibit a strong bias towards texture in visual recognition tasks, human perception is more shape-biased. In this paper, we aim to investigate whether CapsNets, by design, demonstrate a stronger bias toward shape than texture, compared to CNNs. We conducted a series of experiments across multiple capsule architectures on images with a texture-shape cue conflict. Contrary to theoretical expectations, our results show that CapsNets do not consistently exhibit a stronger shape bias than CNNs. Although certain capsule models demonstrate promising shape recognition, they still rely significantly on texture, and their overall performance remains closer to that of CNNs than to human perception. These findings highlight the need for further research and architectural improvements to fully realize the potential of CapsNets in shape-based recognition.

1 Introduction

In recent years, deep learning models have achieved remarkable success across various visual tasks, from image classification to object detection. Convolutional neural networks (CNNs) [16], in particular, have been the cornerstone of many computer vision applications due to their ability to capture local features through convolutional layers. However, Geirhos et al. [9] has revealed that CNNs exhibit a significant *bias toward texture* in visual recognition tasks, often prioritizing fine-grained local details over the global shape of objects. This texture bias has been associated with various limitations in CNNs, including their susceptibility to adversarial attacks and reduced robustness to domain shifts, where object texture may vary but the shape remains consistent.

In contrast, *human visual perception is strongly biased toward shape* rather than texture [2, 7, 22]. Humans can easily recognize an object based on its overall structure and form, even in the presence of changes in texture or fine details. A notable example is the Wadi Sura cave paintings in the Sahara Desert, where ancient artists depicted figures using simple shapes with minimal texture. This demonstrates the human capacity to identify objects through shape alone. Such observations have led to growing interest in developing models that align more closely with human-like shape bias in visual recognition. Figure 1 illustrates how different texture and shape features influence classification.

Capsule Networks (CapsNets), first introduced by Sabour et al. [21], were proposed to overcome the limitations of CNNs, particularly their inability to capture spatial hierarchies and relationships

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

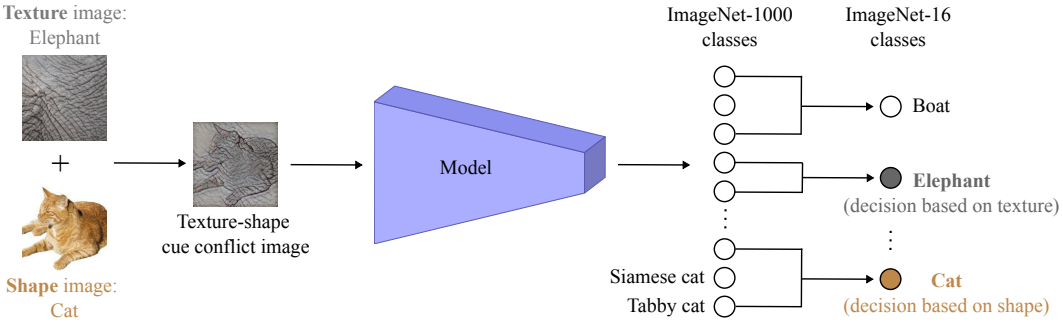

Figure 1: Cue conflict procedure from Geirhos et al. [9] showing a texture of an elephant applied to an image of a cat. The 1000 classes predictions of the models are grouped into 16 basic classes. If the decision is an elephant than the model reflects texture bias, otherwise, if the prediction is a cat, than the model shows shape bias.

between object parts. Capsules are groups of neurons designed to encode the pose and features of the objects in the scene, allowing them to extract more succinct representations with fewer parameters. Capsules in different layers are connected thanks to a dynamic routing mechanism that explicitly models the part-whole or child-parent relationships (e.g., the nose and mouth capsules are children of the face capsule). This design suggests that capsules *may* exhibit a stronger bias towards shape, making them potentially more robust to changes in texture and better aligned with human visual perception.

However, despite these theoretical advantages, the extent to which CapsNets are truly biased toward shape or texture remains an open empirical question. Existing studies have predominantly focused on comparing CNNs and CapsNets regarding performance metrics, such as classification accuracy or robustness, without explicitly measuring how these models handle shape versus texture in visual data. This paper aims to address this gap by providing preliminary empirical evidence to assess whether capsules are indeed more shape-biased or texture-biased. Therefore, we seek to answer the following research question: *Do capsule networks exhibit a stronger shape or texture bias?*

To explore this, we conducted experiments with different CapsNets architectures across several datasets where shape and texture information are systematically manipulated. Our findings reveal that, contrary to the theoretical expectations, CapsNets do not consistently demonstrate a stronger shape bias than CNNs. While some CapsNet models show promising results, their overall reliance on texture remains significant, suggesting that further architectural modifications or training strategies are required to realize their full potential for shape-based recognition.

2 Related work

To provide context for our investigation, in this section, we review the recent findings on the texture-shape bias in neural networks and the development of CapsNets, highlighting key contributions that inform our study.

Shape versus texture bias in neural networks Several researchers tried to investigate and improve generalization towards shapes in neural networks with two main approaches: data augmentation and architectural modification. For the first approach, the pioneering work [9] trained CNNs on Stylized-ImageNet, a stylized version of ImageNet [5] created by applying style transfer techniques to ImageNet images. This encouraged CNNs trained on the dataset to become more shape-biased. Hermann et al. [14] mitigated the texture bias by applying simpler, naturalistic, and human-like augmentation (color distortion, noise, and blur). Their method indicates that differences in how humans and CNNs process images arise from the differences in the data they see. However, other techniques rely not on data augmentation but on architectural improvements. For example, Bahng et al. [1] created a texture-biased model by reducing the receptive field size of CNNs. Others, like Dehghani et al. [4], scaled up vision transformers (ViTs) into 22 billion parameters, showing a near human shape bias performance, and Li et al. [17] introduced shape bias into deep learning models by enforcing sparsity coding constraints.

Capsule networks They were initially introduced by Sabour et al. [21], and since then, several advancements have focused on improving their efficiency and performance. For instance, Ribeiro et al. [20] exploited matrix capsules instead of vector capsules and derived a novel routing algorithm based on Variational Bayes for fitting a mixture of transforming Gaussians, and Edraki et al. [6] modeled entities through capsule subspaces, eliminating the need for any routing mechanism. Moreover, Mazzia et al. [18] replace dynamic routing with self-attention routing to improve generalization, also relying on depthwise convolutions to reduce the number of capsules. Recently, Geng et al. [11] and Renzulli et al. [19] removed redundancy in CapsNet, introducing pruning to reduce computational effort and extract more succinct part-whole relationships, respectively. Additionally, Garau et al. [8], inspired by the theoretical concepts described by Hinton [15], proposed an interpretable model that, compared to previous 'pure' capsule models, brings together CNNs, transformers, neural fields, contrastive learning representation, distillation, and capsules. This model learns better part-whole hierarchies and conceptual-semantic relationships. We have included an overview of the standard CapsNet architecture [21], along with relevant formulas, in Appendix A.1. For a complete review and detailed explanation of CapsNets, please refer to the survey of De Sousa Ribeiro et al. [3].

In contrast to the aforementioned studies, our work, to the best of our knowledge, is the first to empirically investigate the texture-shape bias in CapsNets, providing new insights into how these models handle fundamental visual features.

3 Shape versus texture bias in capsules

3.1 Experiments

Datasets When studying the biases in CapsNets, there are important aspects to consider beyond state-of-the-art performance. Despite their potential advantages, as mentioned in Section 1, CapsNets have not yet seen widespread adoption in the industry or research community, especially compared to CNNs or ViTs. One significant limitation is the lack of standardized architectures and pre-trained models for CapsNets, particularly on large-scale datasets like ImageNet. This challenges researchers and practitioners aiming to benchmark and deploy these models in real-world applications. This gap makes it more difficult to explore architectural improvements or study inherent biases in CapsNets without building models from scratch. As a result, in our experiments, instead of using ImageNet-1000, we trained all models on a subset containing only 224×224 resized images belonging to the 16 categories (we refer to it as ImageNet-16) that are used in the `model-vs-human` toolbox [10]. We used this toolbox to benchmark the gap between humans and CapsNets. Note that ImageNet-16 is highly unbalanced, so we weighted the sampling process by the frequencies of occurrences of each class. Moreover, since there have been no prior results exploring the shape-texture bias in CapsNets, we trained our model without data augmentation to isolate the natural inductive biases of the architecture.

Architectures We tested different variants of capsule models, including different architectures, routing algorithms, and capsule representation (matrix or vector form). We conducted experiments with ResNet-18, standard CapsNets [21] (DR-CapsNets in this paper), Efficient-CapsNets [18], Variational Bayes CapsNets [20] (VB-CapsNets) and Agglomerator [8]. For the first three capsule architectures we extract capsules after three 5×5 convolutional layers with 32, 64, and 128 output channels and a stride of 2. We additionally stacked a DR-CapsNet on top of a ResNet-18 [13] model (ResCapsNet-18). We use the notation `architecture-`L, where L is the number of capsule layers. Note that in every architecture, there are at least two capsule layers (known in the literature as *primary* capsules and *class* capsules. When $L > 4$, we added residual connections between capsules as described by Gugglberger et al. [12]. We employed 16-th (or 4×4 for VB-CapsNets) dimensional capsules in all models. Other hyperparameters, such as the number of routing iterations and training recipes, are the same as presented in the original paper. The code will be publicly available upon acceptance of the paper.

3.2 Results

Figure 2 illustrates the shape and texture bias across different models, including ResNets and various CapsNets, as well as human performance. For the corresponding results on ImageNet-16, please see Appendix A.2.

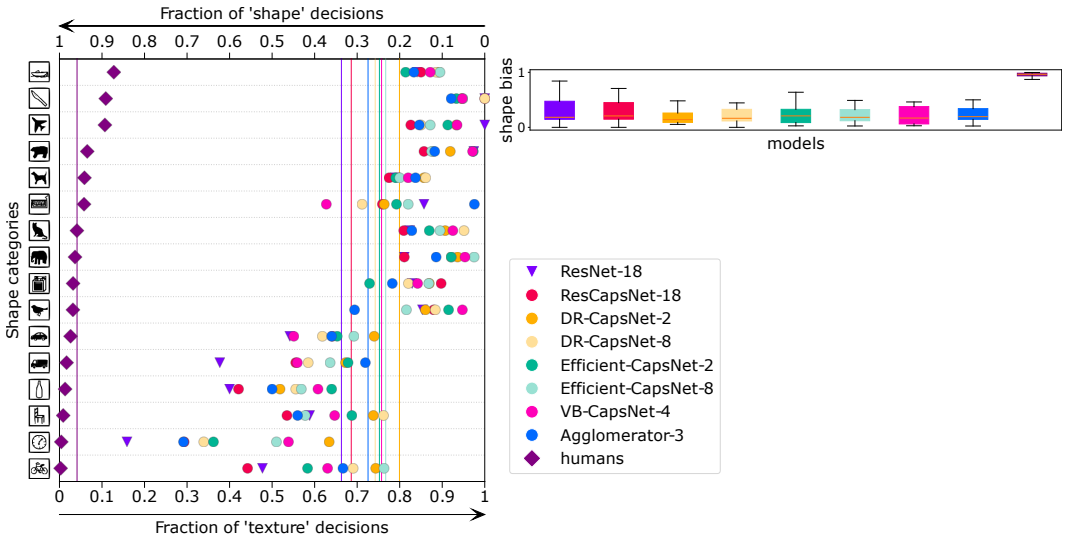

Figure 2: Shape vs. texture biases for stimuli with a texture-shape cue conflict (models trained on ImageNet-16). Solid lines denote the results averaged across all classes.

Several key trends emerge from the figure. Human performance shows significantly less variance in biases across different categories and demonstrates a clear preference for shape-based recognition, as expected from prior research on human visual perception. On the contrary, with deep learning models, this variance is higher.

Among all models, ResNet-18 exhibits the highest shape bias, but its decisions are heavily biased towards textures compared to humans. However, CapsNets, in some cases, avoid the extreme texture bias observed in ResNet-18, particularly in challenging categories like *airplane*, where CapsNets focus more on the object itself than background cues. ResCapsNet-18 and VB-CapsNet are the 'pure' capsule models that demonstrate a shift towards shape-biased decisions, moving closer to ResNet-18 performance, although they still show a high degree of reliance on texture. Interestingly, adding capsules to ResNet-18 even decreases the shape bias compared to the standard ResNet-18. Models such as DR-CapsNet and Efficient-CapsNet exhibit a stronger bias towards textures. We can notice that adding more capsule layers increases the fraction of shape decisions only for DR-CapsNets. In our experiments, with $L > 8$, the training of the models on ImageNet-16 started to diverge.

Agglomerator-3 is the not 'pure' capsule-based model with the closest approximation to human-level shape bias, with a higher fraction of shape-based decisions compared to other CapsNets, suggesting that its architecture may be more effective at capturing the part-whole relationships that facilitate shape recognition.

4 Conclusion

In this study, we investigated the shape-texture bias in capsules and compared their performance to traditional CNNs and human visual recognition. While we hypothesized that CapsNets, due to their architectural innovations such as dynamic routing and part-whole relationships, would exhibit a stronger shape bias than CNNs, the results suggest otherwise. Although some CapsNet models show a shift towards shape-based recognition, their overall performance still demonstrates a noticeable reliance on texture, with most models performing closer to CNNs like ResNet-18 than humans in terms of shape bias.

This outcome challenges the original expectation that CapsNets would inherently prioritize shape over texture, highlighting the need for further exploration and refinement. The variability in shape bias across different CapsNet architectures and depths underscores that the anticipated advantage of capsules is less pronounced than we had hoped.

Future work will address these limitations by exploring alternative methods to enhance the shape bias in CapsNets. One promising approach is to train CapsNets on Stylized-ImageNet [9], a dataset

designed to reduce texture reliance by randomizing textures while preserving object shapes. Additionally, other architectural modifications or data augmentation techniques to improve shape recognition will be considered to align capsules with human-like visual perception better.

References

[1] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 528–539. PMLR, 13–18 Jul 2020.

[2] Irving Biederman and Ginny Ju. Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 20(1):38–64, 1988. ISSN 0010-0285. doi: https://doi.org/10.1016/0010-0285(88)90024-2.

[3] Fabio De Sousa Ribeiro, Kevin Duarte, Miles Everett, Georgios Leontidis, and Mubarak Shah. Object-centric learning with capsule networks: A survey. *ACM Comput. Surv.*, 56(11), jul 2024. ISSN 0360-0300. doi: 10.1145/3674500.

[4] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7480–7512. PMLR, 23–29 Jul 2023.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[6] Marzieh Edraki, Nazanin Rahnavard, and Mubarak Shah. Subspace capsule network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):10745–10753, 2020.

[7] Arlene A. Elder. *Introduction: Writing as Ase*, pages 1–6. Boydell and Brewer, Boydell and Brewer, 2009. ISBN 9781846157479. doi: doi:10.1515/9781846157479-002.

[8] Nicola Garau, Niccolò Bisagno, Zeno Sambugaro, and Nicola Conci. Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13689–13698, 2022.

[9] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[10] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[11] Xinyu Geng, Jiaming Wang, Jiawei Gong, Yuerong Xue, Jun Xu, Fanglin Chen, and Xiaolin Huang. Orthcaps: An orthogonal capsnet with sparse attention routing and pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6037–6046, June 2024.

[12] Josef Gugglberger, David Peer, and Antonio Rodríguez-Sánchez. Training deep capsule networks with residual connections. In Igor Farkaš, Paolo Masulli, Sebastian Otte, and Stefan Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 541–552, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86362-3.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[14] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19000–19015. Curran Associates, Inc., 2020.

[15] Geoffrey Hinton. How to Represent Part-Whole Hierarchies in a Neural Network. *Neural Computation*, 35(3):413–452, 02 2023. ISSN 0899-7667.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[17] Tianqin Li, Ziqi Wen, Yangfan Li, and Tai Sing Lee. Emergence of shape bias in convolutional neural networks through activation sparsity. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[18] Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: capsule network with self-attention routing. *Scientific reports*, 11, 2021.

[19] Riccardo Renzulli, Enzo Tartaglione, and Marco Grangetto. Rem: Routing entropy minimization for capsule networks, 2022.

[20] Fabio De Sousa Ribeiro, Georgios Leontidis, and Stefanos D Kollias. Capsule routing via variational bayes. In *AAAI*, pages 3749–3756, 2020.

[21] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, volume 30, pages 3856–3866. Curran Associates, Inc., 2017.

[22] Johan Wagemans, Joeri De Winter, Hans Op de Beeck, Annemie Ploeger, Tom Beckers, and Peter Vanroose. Identification of everyday objects on the basis of silhouette and outline versions. *Perception*, 37(2):207–244, 2008. doi: 10.1068/p5825. PMID: 18456925.

A Appendix

A.1 Capsule networks background

Here, we describe the fundamental aspects of CapsNets, focusing on the first routing algorithm introduced by Sabour et al. [21], commonly known as dynamic routing (DR-CapsNets).

Capsule Networks (CapsNets) organize neurons into capsules, represented as activity vectors $\boldsymbol{u}$, where each capsule encodes an object or one of its parts. The individual components of these vectors correspond to various properties of the object, such as pose, color, and deformation. This work also refers to the activity vector as the object's pose. The magnitude $\|\boldsymbol{u}\|_2$ of a capsule signifies the probability of the object's presence in the image.

A typical CapsNet consists of multiple capsule layers, denoted by l, stacked on top of a convolutional backbone, where $l \in 1, 2, ..., L$. The capsules in layer l are collectively referred to as $\Omega^{[l]}$. There are three main types of capsule layers: PrimaryCaps (the first layer, built upon convolutional layers), ConvCaps (capsules with localized receptive fields), and FcCaps (fully-connected capsules, where each capsule is connected to all the capsules in the previous layer). Generally, a CapsNet consists of at least two capsule layers: PrimaryCaps and ClassCaps (or FcCaps), with the latter having one output capsule per object class. Figure 3 depicts an example of a general CapsNet architecture.

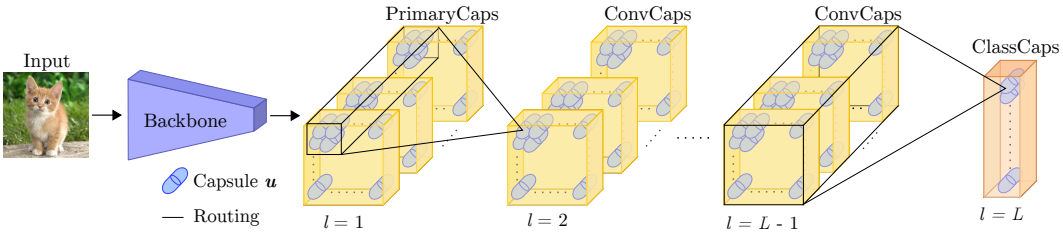

Figure 3: An example of a general CapsNet architecture comprising a convolutional backbone, a PrimaryCaps layer (which is a convolutional layer with squash activation and then reshaping), several ConvCaps layers (where a capsule in layer l is computed using only a subset of capsules in layer $l-1$), and a ClassCaps (or FcCaps, where each capsule in layer l is computed using all the capsules in layer $l-1$) layer.

The process of computing capsules in layer $l+1$ from capsules in layer l involves two stages: part-whole transformation and hierarchical routing. In the first phase, each capsule i in layer l, whose pose is denoted as $\boldsymbol{u}_i^{[l]}$, makes a prediction $\hat{\boldsymbol{u}}_{i,j}^{[l+1]}$, thanks to a transformation matrix $\boldsymbol{W}_{i,j}^{[l]}$, for the pose $\boldsymbol{u}_j^{[l+1]}$ of an upper layer capsule j

$$\hat{\boldsymbol{u}}_{i,j}^{[l+1]} = \boldsymbol{W}_{i,j}^{[l]} \boldsymbol{u}_i^{[l]}. \tag{1}$$

Then, during the second phase, the unnormalized pose $\boldsymbol{s}_j^{[l+1]}$ is computed as the weighted average of votes $\hat{\boldsymbol{u}}_{i,j}^{[l+1]}$

$$\boldsymbol{s}_j^{[l+1]} = \sum_i c_{i,j}^{[l]} \hat{\boldsymbol{u}}_{i,j}^{[l+1]}, \tag{2}$$

where $c_{i,j}^{[l]}$ are the coupling coefficients between a capsule i in layer l and a capsule j in layer $l+1$. The pose $\boldsymbol{u}_j^{[l+1]}$ is then defined as the normalized "squashed" $\boldsymbol{s}_j^{[l+1]}$

$$\boldsymbol{u}_j^{[l+1]} = \text{squash}(\boldsymbol{s}_j^{[l+1]}) = \frac{\|\boldsymbol{s}_j^{[l+1]}\|^2}{1 + \|\boldsymbol{s}_j^{[l+1]}\|^2} \frac{\boldsymbol{s}_j^{[l+1]}}{\|\boldsymbol{s}_j^{[l+1]}\|}, \tag{3}$$

whose magnitude lies in the range [0, 1). The coupling coefficients are computed dynamically and depend on the input. They are determined by a "routing softmax" activation function, whose initial

logits $b_{i,j}^{[l]}$ are the log prior probabilities that the i-th capsule should be coupled to the j-th one

$$c_{i,j}^{[l]} = \text{softmax}(b_{i,j}^{[l]}) = \frac{\exp\left(b_{i,j}^{[l]}\right)}{\sum_k \exp\left(b_{i,k}^{[l]}\right)}. \tag{4}$$

At the first step of the routing algorithm, they are equal, and then they are refined by measuring the agreement (defined as the scalar product) between the pose $\boldsymbol{u}_j^{[l+1]}$ and the prediction $\hat{\boldsymbol{u}}_{i,j}^{[l+1]}$ for a given input. At each iteration, the update rule for the logits is

$$b_{i,j}^{[l]} \leftarrow b_{i,j}^{[l]} + \boldsymbol{u}_j^{[l+1]} \hat{\boldsymbol{u}}_{i,j}^{[l+1]}. \tag{5}$$

The steps defined in Equation 2, Equation 3, Equation 4, Equation 5 are repeated for the r iterations of the routing algorithm. As mentioned in Section 1, the goal of the routing algorithm is to decompose the scene into objects and parts.

Sabour et al. [21] replaced the cross-entropy loss with a *margin loss*: the key idea is that the output capsule for the predicted class should have a long instantiation vector if and only if an object of that class is present in the input image.

A.2 Additional results on ImageNet-16

Table 1: Accuracy and number of trainable parameters of ResNet-18 and different CapsNets on ImageNet-16.

Model	Accuracy	Number of parameters
ResNet-18	0.89	11.2M
ResCapsNet-18	0.91	13.5M
DR-CapsNet-2	0.83	4.9M
DR-CapsNet-8	0.78	5.3M
Efficient-CapsNet-2	0.87	1.7M
Efficient-CapsNet-8	0.84	2.1M
VB-CapsNet-4	0.81	1.4M
Agglomerator-3	0.84	643M