

RobCaps: Evaluating the Robustness of Capsule Networks against Affine Transformations and Adversarial Attacks

Original

RobCaps: Evaluating the Robustness of Capsule Networks against Affine Transformations and Adversarial Attacks / Marchisio, A., De Marco, A., Colucci, A., Martina, M., Shafique, M.. - ELETTRONICO. - (2023), pp. 1-9. (International Joint Conference on Neural Networks (IJCNN) Gold Coast (Australia) 18-23 Giugno 2023) [10.1109/ijcnn54540.2023.10190994].

Availability:

This version is available at: 11583/2987507 since: 2024-04-02T16:47:07Z

Publisher:

IEEE

Published

DOI:10.1109/ijcnn54540.2023.10190994

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

RobCaps: Evaluating the Robustness of Capsule Networks against Affine Transformations and Adversarial Attacks

Alberto Marchisio^{1,*}, Antonio De Marco^{2,*}, Alessio Colucci¹, Maurizio Martina², Muhammad Shafique³

¹Technische Universität Wien, Vienna, Austria ²Politecnico di Torino, Turin, Italy ³New York University, Abu Dhabi, UAE

Email: alberto.marchisio@tuwien.ac.at, s254593@studenti.polito.it, alessio.colucci@tuwien.ac.at

maurizio.martina@polito.it, muhammad.shafique@nyu.edu

Abstract—Capsule Networks (CapsNets) are able to hierarchically preserve the pose relationships between multiple objects for image classification tasks. Other than achieving high accuracy, another relevant factor in deploying CapsNets in safety-critical applications is the robustness against input transformations and malicious adversarial attacks.

In this paper, we systematically analyze and evaluate different factors affecting the robustness of CapsNets, compared to traditional Convolutional Neural Networks (CNNs). Towards a comprehensive comparison, we test two CapsNet models and two CNN models on the MNIST, GTSRB, and CIFAR10 datasets, as well as on the affine-transformed versions of such datasets. With a thorough analysis, we show which properties of these architectures better contribute to increasing the robustness and their limitations. Overall, CapsNets achieve better robustness against adversarial examples and affine transformations, compared to a traditional CNN with a similar number of parameters. Similar conclusions have been derived for deeper versions of CapsNets and CNNs. Moreover, our results unleash a key finding that the dynamic routing does not contribute much to improving the CapsNets’ robustness. Indeed, the main generalization contribution is due to the hierarchical feature learning through capsules.

Index Terms—Machine Learning, Deep Neural Networks, Convolutional Neural Networks, Capsule Networks, Dynamic Routing, Adversarial Attacks, Affine Transformations, Security, Robustness, Vulnerability

I. INTRODUCTION

In recent years, many works have explored the problems of adversarial examples and affine transformations in Convolutional Neural Networks (CNNs) for image classification applications [1] [2] [3] [4]. Szegedy et al. [5] proposed the concept of adversarial examples, i.e., examples with small perturbations, imperceptible to the human eye, that mislead high-confidence models when added to the input. The same limitation of Deep Neural Networks (DNNs) in image classification is also noticed if the input is affected by affine transformations that do not modify the pixels but their relative position in space. The most common means of limiting these problems is to increase the generalization level of a CNN, which is achievable using different methods. Some research works proposed to increase the depth of CNN architectures [6], others proposed to modify the hyper-parameters [7] and using data pre-processing during

the training [8]. For a CNN, the convolutional and the Max Pooling layers provide the generalization and the capability to detect high-order features in a large region of the image (*invariance property*), but without preserving any relation with other identified features.

With the introduction of the Capsule Networks (CapsNets) by Google [9], the basic building block of a neural network, i.e., the neuron, has been replaced by a group of neurons, called *capsule*. The capsules encode spatial information in a vector form. When a detected feature moves around the image, the probability of being detected does not vary, but its *pose* information changes (*equivariance property*). The work in [10] proposes an efficient way of learning the coupling between capsules from different layers through the so-called *dynamic routing* algorithm, an iterative process that replaces the behavior of the max pooling, but without losing any information. Hence, such a capsule structure improves the network’s generalization because it can efficiently learn cross-correlations between different features of the inputs. Recently, Rajasegaran et al. [11] showed that a deeper version of CapsNets can achieve high accuracy also on mid-complex datasets like the CIFAR10 [12], despite reducing the number of parameters compared to the shallower CapsNet in [10].

Existing works [13] [14] [15] [16] have analyzed the vulnerabilities and robustness of CapsNets against affine transformations and adversarial attacks, respectively. *However, they lack a systematic study comparing different types of CapsNets and CNNs and a detailed analysis of the impact of different CapsNet functions (like dynamic routing) on the robustness.* Moreover, Michels et al. [15] did not investigate the CapsNets’ robustness when an adversarial defense, such as the adversarial training [17], is applied.

Such analyses would establish an understanding of differences between CNNs and CapsNets w.r.t. the robustness against adversarial attacks and how the robustness of CapsNets changes depending on the model features. This could help future CapsNet designs in *accounting for the security vulnerabilities into design constraints*, increasing the applicability of CapsNets in real-world scenarios [18].

Research Questions and Associated Challenges

The goal of our paper is to investigate these research questions:

- 1) *Are CapsNets more robust than CNNs against adversarial*

*These authors contributed equally to this work.

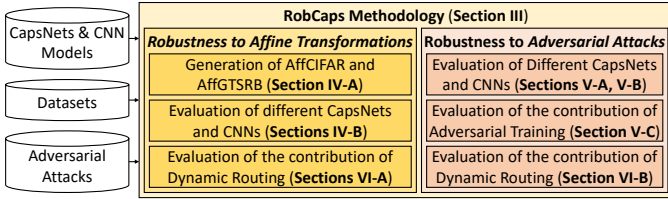


Fig. 1. Overview of our novel contributions in this work.

TABLE I
KEY RESULTS OBTAINED IN THIS PAPER.

		DeepCaps	ResNet20
Affine Transformations Robustness	Accuracy AffNIST	87.60%	96.39%
	Accuracy AffGTSRB	81.14%	89.75%
	Accuracy AffCIFAR	78.66%	75.84%
Adversarial Attacks Robustness	Accuracy MNIST PGD, $\epsilon = 0.05$	1.20%	86.78%
	Accuracy GTSRB PGD, $\epsilon = 0.02$	50.35%	18.99%
	Accuracy CIFAR10 PGD, $\epsilon = 0.005$	37.49%	1.41%

attacks and affine transformations?

- 2) If yes, how can these phenomena be analyzed in a systematic way?
- 3) Which CapsNet functions contribute more to the robustness improvement?

Answering these questions is a challenging task. Firstly, we evaluate a good metric of comparison between CapsNets and CNNs, i.e., which network models give a fair and significant robustness comparison, which types of adversarial attacks are applied, etc. Then, it should be interesting to analyze the transferability of the adversarial attacks, i.e., white-box attacks. *If an adversarial example has been generated to fool network A, does it also fool network B?*

Our Novel Contributions are (see Figure 1):

- We generate an affined-transformed version of the CIFAR10 and GTSRB datasets, called **affCIFAR** and **affGTSRB**, respectively. (Section IV-A)
- We evaluate / compare the **robustness of different CapsNets and CNNs (like ShallowCaps, DeepCaps, ResNet20) against affine transformations** for different datasets and different networks. (Section IV-B)
- We compare the robustness of different networks **against adversarial attacks** for different datasets. Further analyses have been carried out in the presence of a defense such as the **adversarial training**. (Section V)
- We evaluate the role of the **dynamic routing** towards the CapsNets robustness. (Section VI)

In summary, our key results depicted in Table I show that the DeepCaps [11] is more robust than a deeper ResNet20 [6] against affine transformation and different types of adversarial attacks, increasing the complexity of the input data. As we will demonstrate, such improvements in the robustness also hold when the adversarial examples are transferred from one network to the other and vice-versa.

After showing the power of the capsules, we focus our analysis on the dynamic routing, which increases the confidence of the prediction, with a consequent improvement in terms of accuracy. By knowing that, our challenging question is: *Is the dynamic routing also helpful in guaranteeing*

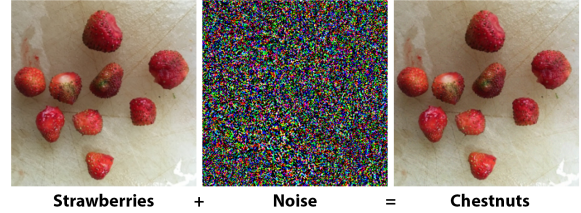


Fig. 2. Example of an adversarial attack's functionality, where strawberries are misclassified as chestnuts [19].

the CapsNets robustness? Our results and analyses provide great insights when relating CNNs and CapsNets against adversarial attacks and affine transformations, as well as how CapsNets' behavior changes when modifying model features.

Before proceeding to the technical sections, we discuss adversarial attacks and CapsNets in Section II to a level of detail necessary to understand our contributions.

II. BACKGROUND AND RELATED WORK

A. Adversarial Attacks

Formally, having an example x that is correctly classified by a well-trained model $M(x) = y_{true}$, an adversarial example $x' = x + \eta$ is defined as a new input, perceptually identical to the original one, but wrongly classified by the model, i.e., $M(x') \neq y_{true}$. Goodfellow et al. [19] proposed the fast gradient sign method (FGSM), a white-box attack to generate adversarial examples by exploiting the gradient of the model w.r.t. the input image, towards the direction of the highest loss. An example of its functionality is shown in Figure 2, where the crafted noise added to the original input is imperceptible to the human eye but results in a misclassification. Afterwards, Madry et al. [17] and Kurakin et al. [20] proposed two different versions of the projected gradient descent (PGD) attack, which is an iterative version of the FGSM that introduces a perturbation α to multiple smaller steps. After each iteration, the PGD projects the generated image into a ball with a radius ϵ , keeping small the size of the perturbation. It is a white-box attack and has both the targeted and untargeted versions. The algorithm consists of the iteration expressed in Equation (1), where θ is the set of parameters and t is the target label.

$$x'_i = x'_{i-1} - \text{proj}_\epsilon(\alpha \cdot \text{sign}(\nabla_x \text{loss}(\theta, x, t))) \quad (1)$$

Carlini and Wagner [21] proposed a powerful white-box targeted attack method that exploits l_∞ , l_1 and l_2 distances to preserve the imperceptibility of the adversarial example. It is performed by solving the optimization problem expressed in Equation (2).

$$\|\eta\|_2 + c \cdot \max(G(x, \eta, t) - M(x, \theta)_t, -k) \quad (2)$$

The algorithm aims to minimize both the components of the equation: (i) the distance η between the input and the adversarial image and (ii) the distance between the max output activation ($G(x, \eta, t) := \max_{i \neq t}(M(x + \eta))$) and the confidence $M(x)_t$ of the target label t . The value c is updated at every iteration to balance the two terms

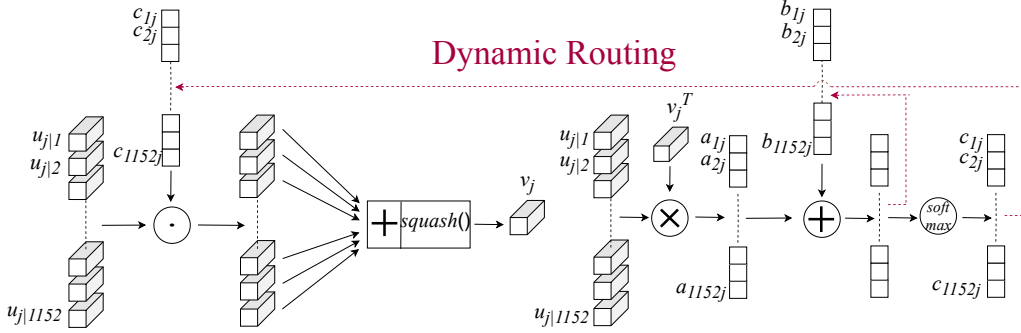


Fig. 3. Schematic overview of the processing flow occurring in the Dynamic routing of the DigitCaps layer.

during the generation of the attacked data. Many works showed the success of such attacks in fooling DNNs and provided state-of-the-art success rate results [19] [21] [17]. A common countermeasure to defend against such attacks is the adversarial training [17], which extends the training set for DNNs by also including the adversarial examples.

B. Capsule Networks

CapsNets gathered attention due to their capability to achieve higher classification accuracy than traditional CNNs. Sabour et al. [10] introduced the first CapsNet architecture, based on the following differences w.r.t. traditional CNNs:

- *capsules*: multi-D entities, instead of single neurons, that constitute each layer.
- a *dynamic routing* algorithm between two adjacent capsules selects the capsules that must be propagated, based on their pose agreement.
- a *squash* function compresses the components of each capsule in a small interval at the end of each layer.

The architecture designed in [10], which we call *ShallowCaps* (for ease of discussion), is composed of:

- a first standard convolutional layer with $256\ 9 \times 9$ kernels.
- a Primary Capsule layer, convolutional with 9×9 kernels and the same parameters as the previous layer, but reshaped to form 32 8-dimensional capsules.
- a DigitCaps layer of 10 capsules of dimension 16.

The last layer defines a transformation matrix that, during the training, learns the relationship between all the capsules of the Primary Capsule layer and the capsules of the DigitCaps layer. The dynamic routing (Fig. 3) has the task of propagating only the activations with a high contribution by updating a set of coupling coefficients. Specifically, this iterative algorithm ensures that only the most voted opinion among the predictions is propagated to the DigitCaps layer.

The limit of this architecture is that it cannot correctly generalize a complex dataset like the CIFAR10. Kumar et al. [22] proposed a three-layer architecture, like the previous one, for the GTSRB dataset [23], increasing the number of capsules coupled with the DigitCaps layer. This one needs a huge number of parameters and wasteful use of resources to reach similar performances as traditional CNN models. To solve this problem, the DeepCaps [11] has been designed to reduce the number of parameters, exploiting deeper capsule

architectures. Without stacking more than one fully-connected layer of capsules, the DeepCaps introduces a new kind of 3D dynamic routing that exploits 3D convolutions.

Both the dynamic routing and the expectation-maximization routing used by Hinton et al. [24] are computationally expensive in terms of execution time. Many works tried to accelerate the procedure at the algorithmic level [25] [26] [27] or at the hardware level [28] [29] [30] [31] [32] [33], and others proposed novel routing strategies [34] [35]. On the contrary, many other works proposed to incorporate the routing procedures into the training process, removing it. In other words, it is possible to learn the coupling coefficients implicitly, including them in the weights of the transformation matrix. Furthermore, [36] proposed a different algorithm introducing new coupling weights between two capsule layers, called *self-routing*.

Our analysis (*Section VI*) also proves that the contribution of the dynamic routing against attacks and affine transformations is not effective. Then, incorporating it into the training process could be a solution to avoid this expensive procedure.

Recent works showed the vulnerability of CapsNet against adversarial attacks. Frosst et al. [37] investigated the detection of adversarial examples using the reconstruction quality of the CapsNets. Peer et al. [38] and Marchisio et al. [39] applied the FGSM method [19] and their proposed attack on CapsNets, respectively. Michels et al. [15] compared the results of different attacks on CapsNets trained on different datasets. The RoHNAS framework [40] includes adversarial robustness among the optimization objectives and conducts Neural Architecture Search to obtain energy efficient and robust CapsNets. However, before employing CapsNets in safety-critical applications, their robustness must be analyzed in practical use-case scenarios, e.g., investigating applications where the CapsNets' classification accuracy is on par or better than the state-of-the-art DNNs, and when robust defenses like adversarial training are adopted.

III. IN-DEPTH VIEW OF OUR ROBCAPS METHODOLOGY

The CapsNets has been considered relatively more robust towards adversarial attacks when compared to traditional CNNs. To investigate this intuition, we present a detailed analysis to answer our main research questions, and to show (1) if and why the Capsule Network under attack provides

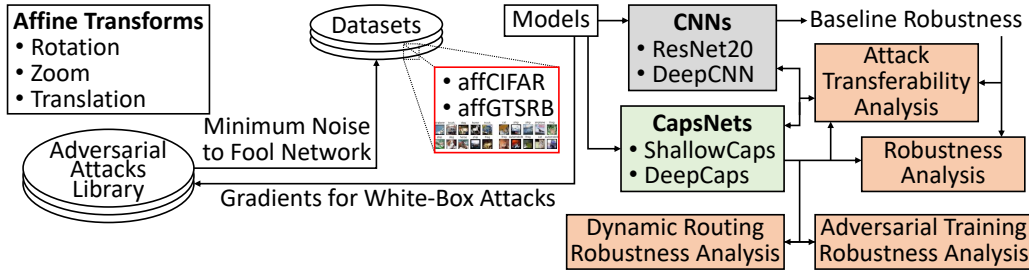


Fig. 4. Overview of our RobCaps methodology.

a better response than traditional CNNs, (2) which model quality plays an important role, and their limits. Knowing the main differences between CapsNets and traditional CNNs, we explore the impact of these networks on affine transformations and adversarial attacks. Moreover, we study the role of different functions of a CapsNet on the robustness against these attacks. Towards a fair and comprehensive evaluation, the results for the ShallowCaps have been compared with three different architectures (chosen according to their properties, their number of parameters, and their depth) for three different datasets, i.e., MNIST [41], GTSRB [23] and CIFAR10 [12].

- A deeper CapsNet architecture, like the *DeepCaps* model [11]. Despite being deeper than the ShallowCaps, it has fewer parameters. The *DeepCaps* employs four groups of 2D convolutional capsule layers, with a 3D convolution layer in the last group and a fully connected capsule layer of 10 32D capsules.
- *ResNet20* (He et al. [6]) is one of the best performing CNN architectures for CIFAR10, used in various applications. It would be interesting to compare the capabilities of the CapsNet with a widely used CNN, which is deeper and employs Residual Blocks with convolutional and average pooling layers.
- A traditional *CNN* with the same depth as the DeepCaps, but without multidimensional entities such as capsules. The dimensions of the layers are reshaped in a 2D fashion, using traditional convolutional layers with batch normalization instead of capsules with squash compression, and a traditional fully connected layer instead of the DigitCaps layer with dynamic routing. Its comparison w.r.t the DeepCaps highlights the contribution to the robustness of 3D convolutions and capsules.

A. Step-By-Step View of our Methodology

Our methodology, shown in Fig. 4, is composed of these following steps:

1) Evaluation of robustness on affine transformations:

- Train our networks with the clean datasets using the same hyperparameters and data augmentation.
- Generate the affine-transformed version of each dataset for a given set of affine-transformations. For the CIFAR10 and the GTSRB datasets, we design two novel transformed datasets with random translations, rotations, and zooms (which we call *affCIFAR* and *affGTSRB*, see Section IV-A).

- Use such *affCIFAR* and *affGTSRB* datasets for inference, as the case for the already existing *affNIST* [42], to evaluate the response of the networks to affine transformations.

2) Evaluation of robustness on adversarial attacks:

We use the saved parameters of the trained models to evaluate the gradient, with respect to the input, for the two implemented white-box attacks. The key steps of our methodology are:

- Apply the projected gradient descent (PGD) attack for each architecture and dataset.
- Test the networks with the generated adversarial inputs, evaluating the accuracy behavior, increasing the perturbation level.
- Apply the Carlini Wagner attack (CW) for each dataset.
- Evaluate the mean distortion required by the algorithm to misclassify 500 images of the test datasets and its fooling rate.
- Apply at the input to a network the adversarial image generated with another one to test the transferability of the attack.
- Test the robustness when the adversarial training defense is applied.

3) Analyzing the contribution of the dynamic routing to the CapsNet’s robustness:

- Modify the dynamic routing of the DigitCaps layer of the DeepCaps and then generate three versions of it with different routing algorithms.
- Analyze the robustness against affine transformations.
- Analyze the robustness against PGD and CW attacks.

B. Experimental Setup

These architectures have been trained with the 40×40 sized version of the MNIST dataset and tested on the *affNIST* for evaluating the robustness against affine transformations. For all the architectures tested on CIFAR10, input data have been resized before the training, from 32×32 to 64×64 , following the pre-processing steps used in [11]. For the GTSRB dataset, the input images’ size is kept at 32×32 . The data augmentation and hyperparameters used for the training are kept the same for all the networks. As a regularization term, the CapsNets have the reconstruction loss provided by the decoder. For the evaluation of the loss, we use the same function as in [10] for CapsNets and the Cross-Entropy for CNNs.

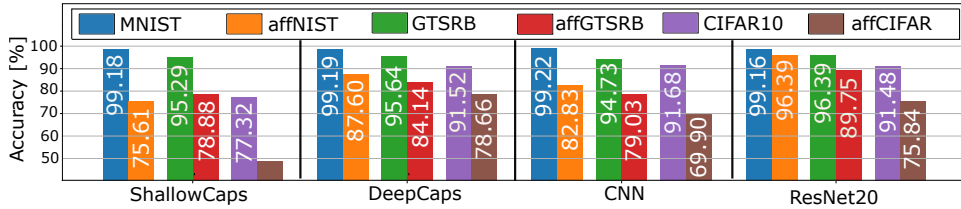


Fig. 5. Robustness against affine transformations.

We implemented the attack algorithms using the CleverHans [43] library, adapted for the Keras framework [44] with Tensorflow backend [45]. The networks have been trained on multiple Nvidia RTX-2080Ti GPUs with CUDA 10. To have a good comparison metric, we train different versions of the DeepCaps architecture modifying/removing the dynamic routing.

IV. ROBUSTNESS AGAINST AFFINE TRASFORMATIONS

A. Affine-CIFAR10 (*affCIFAR*) and Affine-GTSRB (*affGTSRB*) Datasets Generation

While a dataset with affine transformed images of the MNIST dataset (*affMNIST*) is already available, we create an affine version of the CIFAR10 and GTSRB datasets, which we call *affCIFAR* and *affGTSRB*, to compare the response of the networks defined in Section III. The test data was created by modifying the 10 000 test images from the original dataset with random affine transformations. Every image is transformed following these criteria:

- *Translations*: random pixels translations in one or in two dimensions by a factor between 10% and 25% of the input image size, with a fixed interval.
- *Rotations*: random rotations between +20 and -20 degrees with a fixed step.
- *Zooms*: the vertical and horizontal expansions are chosen uniformly between 0.8 (i.e., shrinking the image by 20%) and 1.2 (i.e., enlarging the image by 20%).

B. Affine Trasformations Results

For each model defined in Section III, we evaluate the accuracy for all the datasets and their respective affine-transformed versions. The results are shown in Figure 5.

ShallowCaps vs. DeepCaps: As shown in Figure 5, the ShallowCaps on the CIFAR10 dataset achieves lower accuracy than the state-of-the-art (77.32%). Such limitation is solved by the DeepCaps, which reaches better results even when using the affine version of the respective dataset (78.66%). Thus, using a deeper architecture while keeping the same capsule structure, the DeepCaps model has fewer parameters while having better generalization. Its accuracy with the CIFAR10 dataset (91.52%) and with the affine transformed datasets are much higher compared to ShallowCaps. In fact, despite the shallower model reaching a good accuracy on the normal MNIST and GTSRB datasets, it is still unable to generalize as the DeepCaps against affine transformations. The improvement could also be explained by the presence of the 3D convolutional layer. The effect of having 3D convolutions,

compared to a stack of fully connected capsules, is similar to when we compare the generalization level offered by the Multi-Layer Perceptrons (MLP) and the CNNs. In the DigitCaps layer, each element of the transformation matrix learns if a capsule is correlated with each capsule of the following layer. On the contrary, with the 3D convolution, sliding a 3D kernel, the same weights are used among all the capsules of the layer. This characteristic also allows learning the presence of a particular feature if the input image is spatially transformed (e.g., translated, rotated, or zoomed), preserving the capsule structure.

DeepCaps vs. CNN and ResNet20: Another significant result is provided by comparing the response of the DeepCaps with a traditional CNN, having a similar base architecture. While the accuracy of the CNN on the MNIST, GTSRB, and CIFAR10 datasets is similar to the DeepCaps, the CNN’s robustness against the *affMNIST*, *affGTSRB*, and *affCIFAR* is much lower. These results confirm the benefits of capsules against affine transformations. Compared to the DeepCaps, the ResNet20 is deeper but has fewer parameters. It can generalize better for the *affMNIST* and *affGTSRB* but worse for the *affCIFAR* dataset. This apparently contradictory result is due to the difference in complexity between the datasets. While for simple datasets, a deep CNN, like the ResNet20, can generalize very well, for more complex tasks like the *affCIFAR*, it is outperformed by the DeepCaps. This observation highlights the capability of the capsule architectures to preserve spatial correlations between the features detected, and this difference w.r.t deeper traditional CNNs is even more evident when the input dataset is composed of complex features like the CIFAR10.

V. ROBUSTNESS AGAINST ADVERSARIAL ATTACKS

A. Evaluations under the PGD Attack

We analyze the network response by increasing the perturbation level ϵ , generated by the algorithm. Figures 6a, 6b, and 6c show the results for the MNIST, GTSRB, and CIFAR10 datasets, respectively.

ShallowCaps vs. ResNet20: Applying the PDG attack for the MNIST dataset, the ResNet20 is less vulnerable than other networks for low levels of ϵ . The ShallowCaps robustness behavior, not so far from the one of the ResNet20, overperforms the ResNet20 when $\epsilon \approx 0.065$. Hence, despite the low number of layers, the ShallowCaps responds to the PGD attack similarly to a deeper CNN.

DeepCaps vs. ShallowCaps: According to the results, the ShallowCaps is more robust than the DeepCaps, in contrast

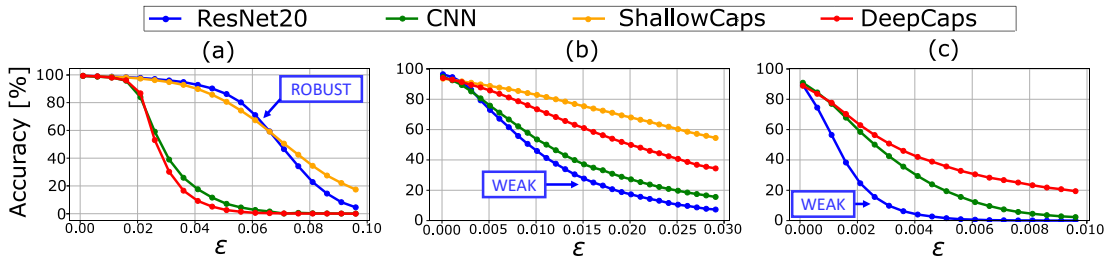


Fig. 6. Robustness against the PGD attack for (a) the MNIST, (b) the GTSRB, and (c) the CIFAR10 datasets.

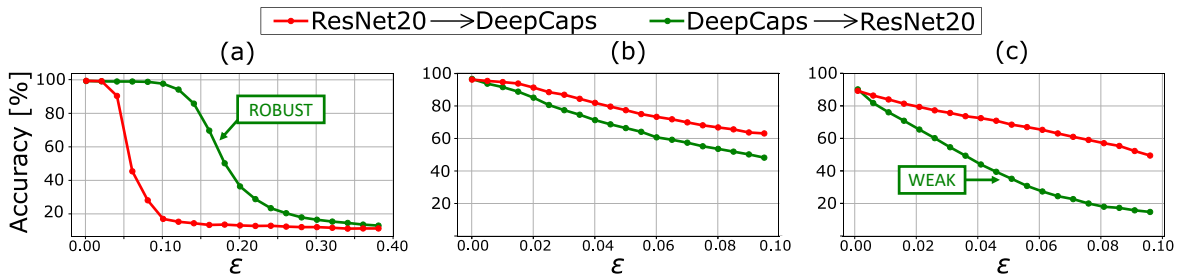


Fig. 7. Transferability for the PGD attack: comparison of the network response for (a) MNIST, (b) GTSRB, and (c) CIFAR10 datasets.

to what happens for affine transformations. This means that increasing the depth of a CapsNet does not provide more robustness against perturbed images. Note that the ShallowCaps response for the CIFAR10 dataset has not been examined because of its very low baseline accuracy, which is not comparable with other networks.

DeepCaps vs. ResNet20 vs. CNN: For this kind of algorithm and the MNIST dataset, Figure 6a shows that the DeepCaps behaves worse than the ResNet20. On the contrary, for more complex datasets like CIFAR10 or GTSRB, the results in Figure 6 show that the ResNet20 is not as robust as for the MNIST dataset. By increasing the perturbation’s size, the attack’s success rate grows faster than on DeepCaps. Such an outcome is in line with the takeaway from Section IV-B, which showed the DeepCaps be more robust than the ResNet20 against the transformations in affCIFAR.

The behavior of the CNN curve for GTSRB and CIFAR10 always stays below the curve of the DeepCaps. It means that the capsule architecture plays a fundamental role in improving the robustness against the PGD attacks when the dataset becomes more complex than the MNIST.

Transferability ResNet20 \iff DeepCaps: Towards a more comprehensive study of the robustness against the PGD, we analyze the transferability of the attacks, between the ResNet20 and the DeepCaps, presenting the two opposite behaviors. We provide as inputs to the DeepCaps the adversarial examples generated with the gradient of the ResNet20 and vice-versa. Figure 7 shows the transferability between these two networks for different datasets.

For the MNIST dataset, the attacks generated for the ResNet20, tested on DeepCaps, have a more significant effect than the opposite way. As shown in Figure 7a, this outcome confirms that the ResNet20 appears suitable for the generalization of the MNIST. The contrasting results can be derived for the GTSRB and CIFAR10 dataset, where the

DeepCaps shows greater robustness than the ResNet20 due to a better generalization ability for a more complex dataset.

B. Evaluations under the Carlini Wagner (CW) Attack

To have a more solid comparison, the CapsNets and CNNs have also been tested against the CW attack, a different kind of algorithm that does not define a threshold for the magnitude of the perturbation (like the ϵ in the PGD attack). It is an iterative targeted algorithm that tries to reduce the gap between the target and the predicted class (success rate) with the minimum perturbation (mean distortion), estimated as the l_2 distance. For a more robust network, the algorithm necessitates more iterations to obtain that the probability of the target class overcomes the probability of the correct class. As a consequence, more iterations also imply a higher l_2 distance between the original image and the adversarial example. For our estimations, we set a maximum of 10 iterations for the MNIST and 5 iterations for the CIFAR10 dataset. In addition, for the attacks on CIFAR10, the algorithm has been forced to set the confidence of the targeted class to 0.5 higher than the confidence of the true label. Table II reports the fooling rate and the mean distortion for both the datasets.

CapsNets vs. CNNs: The CW attack is very effective for traditional CNNs. In fact, it reaches a 100% fooling rate for all three datasets. Similar findings were also made in [21]. On the other hand, both CapsNets show greater robustness (i.e., lower fooling rate) than CNNs, for the MNIST dataset (and also for the GTSRB, even if the fooling rate of the DeepCaps is just a little bit lower than 100%). The CapsNets also require a higher mean distortion than the CNNs, which makes the resulting adversarial example more perceptible. For the CIFAR10 dataset, the CW attack shows its effectiveness because, for all the networks, the fooling rate is 100%. However, we can notice that CapsNets are more robust due to a higher mean distortion.

TABLE II
ROBUSTNESS RESULTS AGAINST THE CW ATTACK.

Network	MNIST		GTSRB		CIFAR10	
	Mean Distortion	Fooling Rate	Mean Distortion	Fooling Rate	Mean Distortion	Fooling Rate
ShallowCaps	1.59	98.6%	1.31	100%	-	-
Deepcaps	1.24	86.8%	1.16	98.8%	0.34	100%
CNN	0.95	100%	0.59	100%	0.23	100%
ResNet20	0.94	100%	0.34	100%	0.12	100%

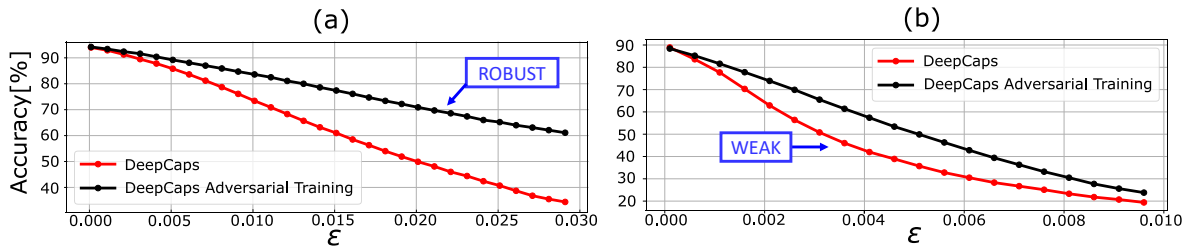


Fig. 8. Adversarially vs. normally trained DeepCaps with (a) the GTSRB and (b) the CIFAR10 datasets.

TABLE III
TRANSFERABILITY OF THE CW ATTACK BETWEEN THE DEEPCAPS AND THE RESNET20.

Network	MNIST	GTSRB	CIFAR10
DeepCaps \rightarrow ResNet20	97.4%	94.0%	86.8%
ResNet20 \rightarrow DeepCaps	97.8%	95.4%	89.2%

DeepCaps vs. ShallowCaps: The DeepCaps appears to be more robust than the ShallowCaps, because of a lower fooling rate, despite having slightly lower mean distortion. Therefore, the depth and the 3D convolutions help to generalize better against the CW attack.

Transferability ResNet20 \iff DeepCaps: Table III shows the transferability of the attacks between ResNet20 and DeepCaps for the CW attack. The values report the accuracies of the two models that receive as input a sample of 500 targeted adversarial examples generated by the CW algorithm applied to the other network. The high accuracy values demonstrate the low level of transferability of the CW attack. Despite this, the ResNet20 still achieves lower accuracies than the DeepCaps, thereby performing less robustly.

C. DeepCaps defended with the PGD Adversarial Training

We also evaluate the robustness of DeepCaps when the PGD adversarial training is applied, compared to the normally trained DeepCaps. We chose an input perturbation ϵ equal to 0.03, with step size 0.005 in each iteration of the algorithm. From Figure 8, we can derive that the adversarial training increases the robustness of the DeepCaps against the PGD attack, both for the CIFAR10 and GTSRB datasets, because its classification accuracy is higher than the baseline DeepCaps.

The adversarial training with PGD defense helps the networks also against the CW attack. For both the datasets, from Table IV, comparing both the mean distortion and the fooling rate, the defended DeepCaps appears more robust. Hence, the adversarial training improves the model interpretability and reduces the learning of brittle features, also when the attack algorithm used for the defense is different from the one used for the actual attack.

VI. ANALYZING THE CONTRIBUTION OF DYNAMIC ROUTING TO THE ROBUSTNESS OF THE DEEPCAPS

As a further analysis, we investigate the contribution of the dynamic routing towards the CapsNets generalization and, as a consequence, towards their robustness. We train two versions of the DeepCaps architecture. (i) The original dynamic routing with three iterations has been replaced by a simple connection (i.e., one iteration of dynamic routing) in both the 3D convolutional and the DigitCaps layers. (ii) The dynamic routing has been replaced by the self-routing algorithm in the last fully connected layer. Then, we run the experiments on such networks and compare them with the original DeepCaps.

A. Evaluations under the Affine Transformations

The results in Table V compare the accuracies achieved by the DeepCaps with and without dynamic routing, and with self-routing, for the MNIST, GTSRB, and CIFAR10 datasets. While the difference is minimal, the response of the DeepCaps without dynamic routing against affine transformations appears to be slightly better. For the CIFAR10 dataset, even if the accuracy with the normal dataset is higher with the dynamic routing compared to the case without it, the latter works better for the affCIFAR dataset. The self-routing shows some limits increasing the complexity of the datasets.

We can derive that the dynamic routing does not contribute significantly to the robustness against affine transformations. Indeed, it makes the DeepCaps much computationally heavier. The functionality of the dynamic routing is to inhibit the propagation of the activation vectors with lower contribution by lowering the values of the coupling coefficients in such connections. Instead, the relationship between objects is learned during training by the transformation matrix, which could wrongly recognize some relationships between the inputs and a wrong output label, which the dynamic routing amplifies, together with the correct agreements. Hence, the confidence of the incorrect label increases.

TABLE IV
ADVERSARIALLY AND NORMALLY TRAINED DEEPCAPS AGAINST THE CW ATTACK.

Network	GTSRB		CIFAR10	
	Mean Distortion	Fooling Rate	Mean Distortion	Fooling Rate
Normally trained DeepCaps	1.16	98.8%	0.34	100%
Adversarially trained DeepCaps	1.44	98.6%	0.84	96.6%

TABLE V
ROBUSTNESS RESULTS AGAINST AFFINE TRANSFORMATIONS.

Network	MNIST40	GTSRB	CIFAR10	AffNIST	AffGTSRB	AffCIFAR
DeepCaps without dynamic routing	99.27%	96.27%	91.47%	87.72%	84.54%	79.86%
DeepCaps with dynamic routing	99.19%	95.29%	91.52%	87.60%	84.14%	78.66%
DeepCaps with self routing	99.25%	95.60%	90.5%	88.15%	83.17%	77.37%

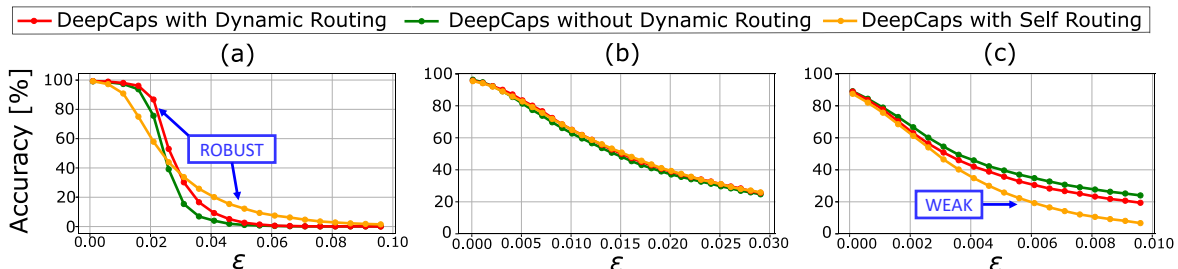


Fig. 9. PGD results: comparison of the DeepCaps response for (a) MNIST and (b) GTSRB and (c) CIFAR10 datasets.

TABLE VI
ROBUSTNESS RESULTS AGAINST THE CW ATTACK.

Network	MNIST		GTSRB		CIFAR10	
	Mean Distortion	Fooling Rate	Mean Distortion	Fooling Rate	Mean Distortion	Fooling Rate
DeepCaps with dynamic routing	1.24	86.8%	1.16	98.8%	0.34	100%
DeepCaps without dynamic routing	1.62	74.0%	1.27	84.11%	0.46	100%
DeepCaps with self routing	2.28	48.6%	1.02	54.4%	0.52	99.2%

B. Evaluations under the Adversarial Attacks

The comparison analysis for the PGD attack applied to the MNIST, GTSRB, and CIFAR10 datasets are shown in Figures 9a, 9b and 9c, respectively.

For the MNIST dataset, the DeepCaps with dynamic routing is slightly more robust than the version without it. On the contrary, for the CIFAR10, the accuracy of the DeepCaps without dynamic routing decreases faster when increasing the perturbation ϵ . We can conclude that increasing the complexity of the dataset, from MNIST toward the CIFAR10, the dynamic routing does not improve the classification capability when the input starts to be perturbed.

Table VI shows the results of the CW attack. The self-routing seems to confer robustness with this attack, even if the architecture with dynamic routing is again outperformed by the one without it. Since the fooling rate is lower and the mean distortion is higher without dynamic routing, we can derive that the dynamic routing does not improve the robustness against such an attack. It confirms that the dynamic routing does not contribute much to the generalization.

VII. CONCLUSION

In this paper, we proposed a methodology to systematically analyze the robustness of CapsNets against affine

transformations and adversarial attacks. Comparing CapsNets and CNNs, we investigated which differences play critical roles in increasing the robustness. The ShallowCaps are more robust than comparable CNNs. However, despite the high cost of training many parameters, they do not generalize well on more complex datasets. The analysis results demonstrate that they are more robust against adversarial attacks but show their limits against affine transformations. Going deeper, the DeepCaps model reduces this problem, decreasing the gap between the transformed and untransformed versions of the datasets, despite the lower number of parameters. Against the adversarial attacks, the DeepCaps does not reach the same robustness as the ShallowCaps for a simple task like the MNIST classification. However, for a more complex dataset like the CIFAR10, their performances overcome not only a CNN with a similar architecture but also the ResNet20. In addition, the DeepCaps offers even higher robustness when the adversarial training is employed. The same conclusion can be obtained for the affine transformations, where the DeepCaps reaches a higher accuracy than the ResNet20 with the affCIFAR dataset. Moreover, our results show that the dynamic routing does not contribute much to improving the CapsNets' robustness.

Our thorough analysis paves the way for future CapsNet

designs, allowing designers to take into account adversarial attacks when targeting safety-critical applications, as well as leading the path for new adversarial attacks against CapsNets.

ACKNOWLEDGMENT

This work has been supported in part by the Doctoral College Resilient Embedded Systems, which is run jointly by the TU Wien's Faculty of Informatics and the UAS Technikum Wien. This work was also supported in parts by the NYUAD Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010, the NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001, and the NYUAD Center for CyberSecurity (CCS), funded by Tamkeen under the NYUAD Research Institute Award G1104.

REFERENCES

- [1] M. Capra, B. Bussolino, A. Marchisio, G. Masera, M. Martina, and M. Shafique, "Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead," *IEEE Access*, vol. 8, pp. 225134–225180, 2020.
- [2] A. Marchisio, M. A. Hanif, F. Khalid, G. Plastiras, C. Kyrkou, T. Theoharides, and M. Shafique, "Deep learning for edge computing: Current trends, cross-layer optimizations, and open research challenges," in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2019.
- [3] M. Capra, B. Bussolino, A. Marchisio, M. Shafique, G. Masera, and M. Martina, "An updated survey of efficient hardware architectures for accelerating deep convolutional neural networks," *Future Internet*, 2020.
- [4] S. Dave, A. Marchisio, M. A. Hanif, A. Guesmi, A. Shrivastava, I. Alouani, and M. Shafique, "Special session: Towards an agile design methodology for efficient, reliable, and secure ML systems," in *40th IEEE VLSI Test Symposium (VTS)*, pp. 1–14, IEEE, 2022.
- [5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations (ICLR)*, 2014.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [7] S. R. Young, D. C. Rose, T. P. Karnowski, S. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," in *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments (MLHPC)*, pp. 4:1–4:5, ACM, 2015.
- [8] W. Zhang, Y. Kinoshita, and H. Kiya, "Image-enhancement-based data augmentation for improving deep learning in image classification problem," in *IEEE International Conference on Consumer Electronics (ICCE-TW)*, 2020.
- [9] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *International Conference on Artificial Neural Networks (ICANN)*, pp. 44–51, Springer, 2011.
- [10] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [11] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "Deepcaps: Going deeper with capsule networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [12] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.
- [13] J. Gu and V. Tresp, "Improving the robustness of capsule networks to image affine transformations," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7283–7291, 2020.
- [14] A. Marchisio, G. Nanfa, F. Khalid, M. A. Hanif, M. Martina, and M. Shafique, "Sevuc: A study on the security vulnerabilities of capsule networks against adversarial attacks," *Microprocessors and Microsystems*, 2023.
- [15] F. Michels, T. Uelwer, E. Upschulte, and S. Harmeling, "On the vulnerability of capsule networks to adversarial attacks," *CoRR*, vol. abs/1906.03612, 2019.
- [16] A. Marchisio, G. Caramia, M. Martina, and M. Shafique, "fakeweather: Adversarial attacks for deep neural networks emulating weather conditions on the camera lens of autonomous systems," in *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2022.
- [17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations (ICLR)*, 2018.
- [18] M. Shafique, A. Marchisio, R. V. W. Putra, and M. A. Hanif, "Towards energy-efficient and secure edge AI: A cross-layer framework ICCAD special session paper," in *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9, IEEE, 2021.
- [19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations (ICLR)*, 2015.
- [20] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations (ICLR)*, 2017.
- [21] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.
- [22] A. D. Kumar, "Novel deep learning model for traffic sign detection using capsule networks," *CoRR*, vol. abs/1805.04424, 2018.
- [23] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2013.
- [24] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *6th International Conference on Learning Representations (ICLR)*, 2018.
- [25] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, and S. Zhang, "Investigating capsule networks with dynamic routing for text classification," in *Conference on Empirical Methods in Natural Language Processing*, 2018.
- [26] A. Marchisio, B. Bussolino, A. Colucci, M. A. Hanif, M. Martina, G. Masera, and M. Shafique, "Fastcaps: An integrated framework for fast yet accurate training of capsule networks," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.
- [27] A. Marchisio, A. Massa, V. Mrazek, B. Bussolino, M. Martina, and M. Shafique, "Nascaps: A framework for neural architecture search to optimize the accuracy and hardware efficiency of convolutional capsule networks," in *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 114:1–114:9, IEEE, 2020.
- [28] A. Marchisio, M. A. Hanif, and M. Shafique, "Capsacc: An efficient hardware accelerator for capsulenets with data reuse," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 964–967, IEEE, 2019.
- [29] A. Marchisio, B. Bussolino, A. Colucci, M. Martina, G. Masera, and M. Shafique, "Q-capsnets: A specialized framework for quantizing capsule networks," in *ACM/IEEE Design Automation Conference (DAC)*, 2020.
- [30] A. Marchisio, V. Mrazek, M. A. Hanif, and M. Shafique, "Red-cane: A systematic methodology for resilience analysis and design of capsule networks under approximations," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1205–1210, IEEE, 2020.
- [31] A. Marchisio, B. Bussolino, E. Salvati, M. Martina, G. Masera, and M. Shafique, "Enabling capsule networks at the edge through approximate softmax and squash operations," in *ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 27:1–27:6, ACM, 2022.
- [32] A. Marchisio, V. Mrazek, M. A. Hanif, and M. Shafique, "FECCA: design space exploration for low-latency and energy-efficient capsule network accelerators," *IEEE Trans. Very Large Scale Integr. Syst. (TVLSI)*, 2021.
- [33] A. Marchisio, V. Mrazek, M. A. Hanif, and M. Shafique, "Descnet: Developing efficient scratchpad memories for capsule network hardware," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD)*, 2021.
- [34] J. Choi, H. Seo, S. Im, and M. Kang, "Attention routing between capsules," in *2019 IEEE/CVF International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1981–1989, IEEE, 2019.
- [35] H. Li, X. Guo, B. Dai, W. Ouyang, and X. Wang, "Neural network encapsulation," in *European Conference on Computer Vision (ECCV)*, pp. 266–282, 2018.
- [36] T. Hahn, M. Pyeon, and G. Kim, "Self-routing capsule networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [37] N. Frosst, S. Sabour, and G. E. Hinton, "DARCCC: detecting adversaries by reconstruction from class conditional capsules," *CoRR*, vol. abs/1811.06969, 2018.
- [38] D. Peer, S. Stabinger, and A. J. Rodríguez-Sánchez, "Training deep capsule networks," *CoRR*, vol. abs/1812.09707, 2018.
- [39] A. Marchisio, G. Nanfa, F. Khalid, M. A. Hanif, M. Martina, and M. Shafique, "Capsattacks: Robust and imperceptible adversarial attacks on capsule networks," *CoRR*, vol. abs/1901.09878, 2019.
- [40] A. Marchisio, V. Mrazek, A. Massa, B. Bussolino, M. Martina, and M. Shafique, "Rohnas: A neural architecture search framework with conjoint optimization for adversarial robustness and hardware efficiency of convolutional and capsule networks," *IEEE Access*, 2022.
- [41] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, 2012.
- [42] T. Tieleman, "The affnist dataset," *cs.toronto.edu*, 2013.
- [43] I. J. Goodfellow, N. Papernot, and P. D. McDaniel, "cleverhans v0.1: an adversarial machine learning library," *CoRR*, vol. abs/1610.00768, 2016.
- [44] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.
- [45] M. Abadi et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016.