GERALT: Real-time Detection of Evasion Attacks in Deep Learning Systems

Xiangru Chen and Sandip Ray
Department of ECE, University of Florida, Gainesville, FL 32611. USA
cxr1994816@ufl.edu, sandip@ece.ufl.edu

2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10181921

***Abstract*—Evasion attacks constitute an important class of security vulnerabilities in deep learning systems. In this attack, the adversary can coerce the victim DNN into targeted misclassification with slightly modified input data. While detection and defense methods have been proposed for evasion attacks, they incur high overhead and cannot be employed for real time detection on resource-constrained devices. In this paper, we propose an infrastructure, GERALT, to optimize evasion attack detection for real-time execution. GERALT includes a software component to optimize detection methods enabling the use of a smaller detection network, and hardware architecture to accelerate inter-network inference with intermediate data reuse techniques. Our evaluation demonstrates that GERALT achieves more than 3x improvement in performance over standard accelerators like Eyeriss without affecting detection and classification accuracy.**

***Index Terms*—Evasion Attack Detection, Accelerator, Deep Neural Network, Adversarial Example, Image Classification**

I. INTRODUCTION

A quintessential DNN application is *image classification*, which entails associating one (single-label classification) or more (multi-label classification) labels to a given image. DNNs have been successfully used for classifying images from a variety of domains including handwritten digits and characters[1], [25], 3D toys [15], human and animal faces [20], etc. Image classification is also critical to the development of autonomous driving technology, which depends on Computer Vision systems typically realized through DNNs for perception of environment. However, a critical limitation in the applicability of DNNs in image classification is their vulnerability to *evasion attack* [24], [19]. This attack perturbs the input image with noises targeted towards inducing misprediction. These noise-augmented images, — also referred to as *adversarial examples*, — can subvert the efficacy or even viability of classification systems in critical applications. For example, in an automotive Computer Vision system, an adversarial example of a perturbed stop sign could be mispredicted as a different road sign resulting in a catastrophic accident.

There has been significant work on detecting evasion attacks [17], [4], [27]. Most of these works utilize machine learning techniques for detection with high accuracy. However, in spite of their effectiveness, it is infeasible to employ them as a real-time detection strategy for evasion attacks on deep learning systems because of the overhead of inference computation. In particular, applying the detection techniques incurs more than two times the computation cost of normal inference[2].

In this paper, we propose a mechanism for extending evasion attack detection neural network for real-time execution. Our framework, GERALT includes a combination of software module (GERALT-func) and hardware architecture (GERALT-arch) to accelerate inter-networking inference and provide architecture support for optimizing detection neural networks. GERALT-func optimizes the computation efficiency of detection network (DN) to use smaller neural network for detection and bypass the previous layers of classification network (CN) with intermediate data. The approach is inspired by Transfer Learning, and can be leveraged effectively to reduce inter-network computation. GERALT-arch accelerates the inter-network computation with intermediate data reuse.

The paper makes the following important contributions.

- GERALT represents a unique software-hardware design to enable real-time detection of evasion attacks.
- GERALT-func is developed to optimize the computation efficiency of detection method to fit in the edge devices without loss of detection and classification accuracy.
- GERALT-arch represents the first inter-network inference accelerator to reuse intermediate data and reduce computation with special storage pattern and dataflow.
- We demonstrate the effectiveness of GERALT in optimization of Feature Squeezing [27]. Feature Squeezing is a promising approach for detecting evasion attacks. However, it incurs significant computation overhead making it infeasible for real-time detection. Our evaluation shows how to use GERALT for systematic optimization of Feature Squeezing to enable real-time detection.

II. RELATED WORK

Evasion attack entails an adversary perturbing images of a class to coerce an image classification system to erroneously mis-label them as members of a different class. Recent demonstrations have shown that even a small perturbation can result in a significant reduction in accuracy of DNNs [24], [19], [16], [22], [26]. There has been significant research on detection and defense of evasion attack. SafetyNet [17] computes the appearance of abnormal relu results to reject attacked inputs. Carrara *et al.* [4] train an LSTM to learn the transfer of input representatives, which map to different areas for attacked images. A detector Neural Network [18] is attached to a large network to be trained together to improve the detecting accuracy adaptively. Feature Squeezing [27] coalesces input

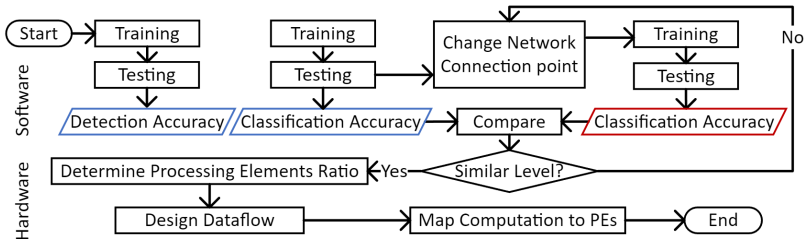


Fig. 1: GERALT Design Flow

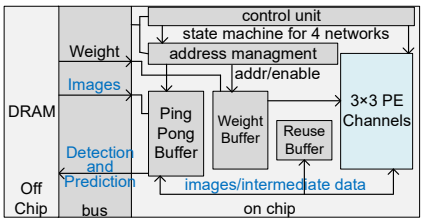


Fig. 2: GERALT-arch Overview

samples that correspond to many different feature vectors in the original space of the input sample and utilizes differences between the predictions of classification DNN itself with squeezed inputs to detect attacks. The comparison between detection methods is discussed in [2].

Many accelerator designs has been proposed with remarkable performance. Equinox [7] supports interleave training between idle inferences to improve utilization. PUMA [3] is designed for scalable computation of different neural network models with high energy efficiency using advanced memristor. The inter-network training has been explored in Dandelion [5] with shared data and computation resources. GCONV [28] manages to map different computations to the same pattern to aid inference or training accelerator design. However, unlike GERALT, these approaches(together with GPU [21], TPU [12] and other architectures [11], [13], [10]) do not utilize the opportunity in inter-network inference.

III. GERALT OVERVIEW

A key challenge in real-time detection using neural network is its additional computational overhead. Note that a naive application of a detection neural network requires about two times the computation of normal inference. GERALT addresses this problem through a combination of software optimization and hardware support. Fig 1 outlines the flow of our design.

A. GERALT-*func: Optimizing Detection Network*

GERALT-func targets optimization of adversarial example detection by separating the computation involved in detecting the adversarial examples from the classification computation. The deep learning system can then utilize a detection network while transmitting the intermediate data to the classification network for further computation to reduce overhead.

There are several challenges involved in realizing the approach. First, since another neural network is used for detection, the size of intermediate feature map may not be suitable to be loaded by classification network for inference. Furthermore, even when two networks share the same size of feature map, the number of channels varies between layers. Finally, it is hard to decide which layer contains the most useful information to be used for classification. To address these challenges and design the inter-network structure with similar performance to the original model (classification and detection), GERALT-func involves the following steps.

Collecting base accuracy. A standard neural network is chosen as the baseline of classification accuracy. Another neural network is trained as the baseline detection network. Adversarial attack is applied to the dataset to get adversarial examples. Detection network is evaluated on the data comprising a combination of original and adversarial images to get the basic detection accuracy.

Exploring the best inter-network inference structure. The following two options are utilized to connect two neural networks. The results are compared to decide the best configuration of GERALT-func (See Section IV).

1) Dimension modification is applied to the detection model structure to provide the classification model with suitable size of feature map from different layers.
2) Only Feature maps are resized and forwarded to the classification model.

B. GERALT-*arch: Accelerating Inter-Network Inference*

Although GERALT-func optimizes evasion attack detection by using a smaller detection network, there is no corresponding architecture support from current accelerators. GERALT-arch addresses this by providing inter-network design for intermediate data reuse. The GERALT-arch design is shown in Fig. 2, which leverages three critical observations. First, unlike Neural Network training, it is not necessary to store intermediate data for following computation (back propagation and weight update). This reduces the requirement of off-chip storage and bandwidth, allowing smaller DRAM with less energy consumption. Second, the computation time for one image is relatively short and the image loading becomes more frequent. This increases the threshold of on-chip latency. Finally, the reuse of intermediate data includes additional 'write' operations from detection network rather than only 'read' operations from classification network. This implies that pipelining the computation of detection and the classification networks would result in different operations occurring sequentially within the same memory space.

1) Storage Pattern: Considering the design choices above, we assemble the storage architecture for GERALT-arch in Fig. 3. There are four groups of ping-pong buffers to store the computation results, three for detection and one for classification. Two groups of weight buffers are used to store all kernel weights: one is shared between three smaller detection networks and one is for classification network. A group of special reuse buffers is deployed to store the intermediate

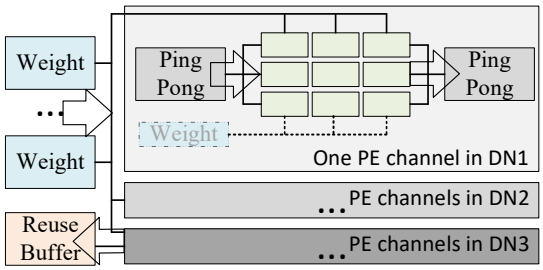


Fig. 3: GERALT Storage: There is only 1 group of DN1's Ping-pong buffer in this figure. Ping-Pong buffers of DN2,3 and CN are not shown.

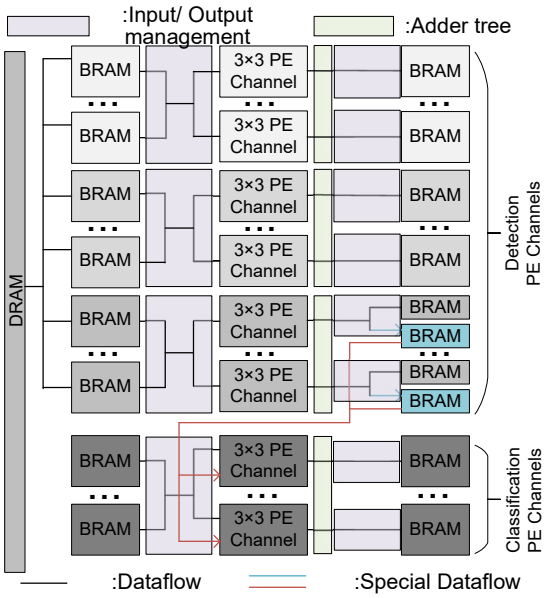


Fig. 4: GERALT Dataflow

data of one certain layer, which supports the pipeline between networks.

2) Dataflow of Inter-Network Inference: The data flow modifications are highlighted in Fig. 4. The image is loaded from off-chip memory to ping-pong buffers of three detection networks. They perform inference of front layers in stationary weight until reaching the connection point. The outputs of connection point are stored to the reuse buffer in the same speed as for ping-pong buffer to avoid idleness. The detection networks then reach the next connection point and provide new data for computation to build the pipeline (Fig. 5). The classification computation starts from the data in the reuse buffers with fixed address pattern as same as reading data from ping-pang buffers. For optimal latency, the time for classification computation should be equal to that for detection. Thus, we determine the ratio of PE channels as 63:1.

IV. GERALT EVALUATION

A. Evaluation of GERALT-func

AlexNet [14] and ResNet-50 [8] are trained on the traffic dataset containing 51,839 images of 43 common classes of traffic sign[23] as the classification network. SqueezeNet [9] and ResNet-18 are utilized for evasion attack detection. Deepfool [19] attack is applied to get adversarial examples.

Table I and Table II show the results of configuration experiments. The settings of these configuration are selected to maximize the classification accuracy. Here "Connection point – SQZ" means that the outputs of the layer in SqueezeNet are reused and "Connection point – AN" means that this layer in AlexNet takes the intermediate data as input. "Connection point – 18" and "Connection point – 50" indicate the same meanings for experiments using ResNet. "Resize" and "Structure" indicate the methods to match the size between networks mentioned in Section III-A. Experiments are restricted between the layers sharing the same number of channels.

Although the best accuracy of "Resize" method drops 13%, we still get several interesting observations of 'Resize' method in the first 12 settings of Table I. First, the accuracy is determined by the information contained in "Connection point-SQZ". Even when the "Connection point – AN" is changed, the accuracy varies a little. Second, the accuracy is affected by the information suitability of "Connection point - AN": the latter layer performs better as the "Connection point – AN", shown in setting 4, 7 and 10. Finally, setting 17 has the best accuracy, which implies that the fire5 layer of SqueezeNet contains most useful information for final prediction and the last few layers of AlexNet can fit this information better.

From the experiments we can conclude that changing the network structure outperforms resizing feature map directly, which results from the lost information during resizing operation and the mismatch between connection points. These results enable the optimization of computation between any two CNNs. Based on previous results, GERALT uses setting 17 as the functional configuration. And Table II further proves the generalization of connection-point.

Our evaluation also includes the detection accuracy after optimization of computation with GERALT-func. The detection accuracy of optimized Feature Squeezing is 0.5% higher than traditional one with 94.8% of truth positive rate.

B. Evaluation of GERALT-arch

GERALT-arch is implemented on Xilinx VCU118 evaluation board, equipped with 76Mb BRAM on chip and 20Gb DDR4 off chip. We deployed 64 3×3 PE channels. Each ping-pong buffer has the size of 0.15Mb for detection PE channels, and they share 21 weight buffers with 0.24 Mb weights. The baseline1 is the optimized weight stationary architecture used in Dandelion [5] to deal with large variance of kernel size while using PE channel hardware configuration of the same scale with GERALT-arch. The popular inference accelerator, Eyeriss [6] is the baseline2. The baselines are used in two modes: (1) naively running 3 classification networks sequentially to

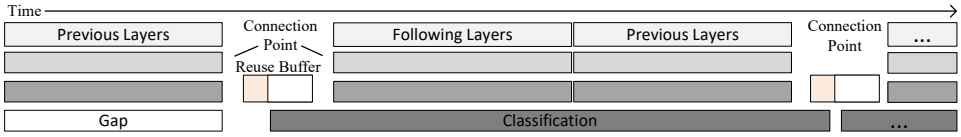
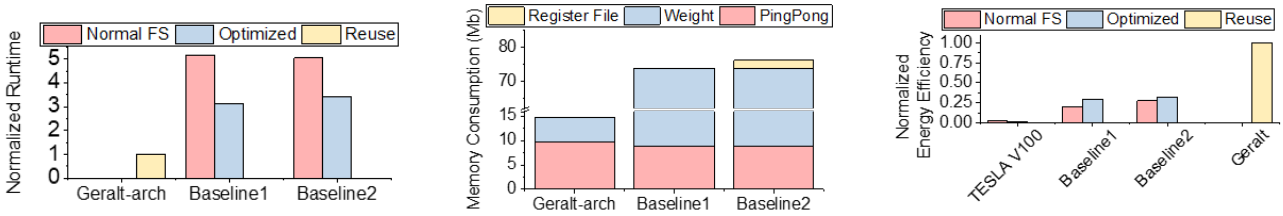


Fig. 5: GERALT Pipeline



(a) (b) (c)

Fig. 6: GERALT-arch Evaluation. (a) Normalized Run-time; (b) Memory Consumption; (c) Energy Efficiency

TABLE I: Accuracy of AlexNet Connected with SqueezeNet

Setting	Connection Point-SQZ	Connection Point-AN	Resize	Structure	Accuracy
Origin	N/A	N/A	N/A	N/A	93.63
0	fire6	relu3	Yes	No	62.29
1	fire7	relu3	Yes	No	62.29
2	fire6	conv4	Yes	No	62.29
3	fire7	conv4	Yes	No	62.29
4	fire4	relu4	Yes	No	60
5	pool2	relu4	Yes	No	32.57
6	fire5	relu4	Yes	No	56.57
7	fire4	conv5	Yes	No	60
8	pool2	conv5	Yes	No	32.57
9	fire5	conv5	Yes	No	56.57
10	fire4	relu5	Yes	No	76.57
11	pool2	relu5	Yes	No	32.57
12	fire5	relu5	Yes	No	68.57
13	fire4	relu5	No	Yes	87.43
14	fire5	relu5	No	Yes	89.14
15	fire4	conv5	No	Yes	91.20
16	pool2	conv5	No	Yes	94.23
17	fire5	conv5	No	Yes	94.70

TABLE II: Accuracy of ResNet-50 Connected with ResNet-18

Setting	Connection Point-18	Connection Point-50	Resize	Structure	Accuracy
Origin	N/A	N/A	N/A	N/A	96.05
0	conv1x	conv2x	No	Yes	95.92
1	conv1x	conv3x	No	Yes	96.29
2	conv1x	conv4x	No	Yes	96.37

get the detection and prediction together; and (2) running optimized Feature Squeezing method without intermediate data reuse.

Fig. 6(a) shows the run time improvement of GERALT-arch. For optimized Feature Squeezing, using smaller network for detection greatly reduces the computation requirement. Furthermore, reusing intermediate data further improves the performance with three times speedup. The computations of classification's previous layers are skipped which takes large part of run-time in baseline. The memory usage also varies as shown in Fig. 6(b). Although GERALT-arch occupies more BRAM space for intermediate data because of the pipeline execution, most storage of the kernel weights of big classification network can be avoided. Also, the 50 times larger of the AlexNet model results in the huge gap of weight buffer size. Finally, we compute the energy efficiency of each architecture in Fig. 6(c) to consider their viability on edge devices. Since the GPU computation is hard to scale down, we only calculate its energy efficiency which shows the higher power and under-utilization when dealing with less dense computation. GERALT incurs less off-chip DRAM accesses which benefit from less and fully buffered weights. Second, the smaller size of on-chip BRAM and less required register files contribute to higher efficiency.

V. CONCLUSION

We proposed an infrastructure, GERALT, as the optimization and architecture support for enabling real-time detection of evasion attacks. GERALT comprises a software component: GERALT-func to optimizes detection neural network method to reduce computation without compromising accuracy, and GERALT-arch to enable inter-network acceleration with the reuse of intermediate data. Our experimental results show that GERALT achieves significant improvement on performance with high energy efficiency and low memory consumption.

In future work, we will perform more experiments on GERALT and study the principle of inter-network system design. We will also explore the extension of GERALT for optimizing other strategies for evasion attack detection and defense, *e.g.*, denoising and GAN.

Acknowledgements: This research has been partially supported by the Semiconductor Research Corporation under Contract 2021-HWS-3060 and National Science Foundation under Grant SATC-2221900.

REFERENCES

[1] M. M. Abu Ghosh and A. Y. Maghari. A comparative study on handwriting digit recognition using neural networks. In *2017 International Conference on Promising Electronic Technologies (ICPET)*, pages 77–81, 2017.

[2] A. Aldahdooh, W. Hamidouche, S. A. Fezza, and O. Déforges. Adversarial example detection for dnn models: A review and experimental comparison. *Artificial Intelligence Review*, pages 1–60, 2022.

[3] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, G. Ndu, M. Foltin, R. S. Williams, P. Faraboschi, W.-m. W. Hwu, J. P. Strachan, K. Roy, and D. S. Milojicic. Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19, page 715–731, New York, NY, USA, 2019. Association for Computing Machinery.

[4] F. Carrara, R. Becarelli, R. Caldelli, F. Falchi, and G. Amato. Adversarial examples detection in features distance spaces. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[5] X. Chen, J. Zhang, and S. Ray. Dandelion: Boosting dnn usability under dataset scarcity. *IEEE Transactions on Computers*, pages 1–1, 2021.

[6] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, 52(1):127–138, 2017.

[7] M. Drumond, L. Coulon, A. Pourhabibi, A. C. Yüzügüler, B. Falsafi, and M. Jaggi. Equinox: Training (for free) on a custom inference accelerator. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, page 421–433, New York, NY, USA, 2021. Association for Computing Machinery.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[10] H. Jia, M. Ozatay, Y. Tang, H. Valavi, R. Pathak, J. Lee, and N. Verma. 15.1 a programmable neural-network inference accelerator based on scalable in-memory computing. In *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, volume 64, pages 236–238, 2021.

[11] J. Jo, S. Cha, D. Rho, and I.-C. Park. Dsip: A scalable inference accelerator for convolutional neural networks. *IEEE Journal of Solid-State Circuits*, 53(2):605–618, 2018.

[12] N. P. Jouppi, C. Young, and Patil. In-datacenter performance analysis of a tensor processing unit. *SIGARCH Comput. Archit. News*, 45(2):1–12, jun 2017.

[13] J. H. Kim, B. Grady, R. Lian, J. Brothers, and J. H. Anderson. Fpga-based cnn inference accelerator synthesized from multi-threaded c software. In *2017 30th IEEE International System-on-Chip Conference (SOCC)*, pages 268–273, 2017.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[15] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104 Vol.2, 2004.

[16] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[17] J. Lu, T. Issaranon, and D. Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[18] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[19] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth. Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2. *Sustainable Cities and Society*, 66:102692, 2021.

[21] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899, 2008.

[22] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[23] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012.

[24] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[25] C. Wu, W. Fan, Y. He, J. Sun, and S. Naoi. Handwritten character recognition by alternately trained relaxation convolutional neural network. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 291–296, 2014.

[26] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

[27] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[28] J. Zhang, X. Chen, and S. Ray. Gconv chain: Optimizing the whole-life cost in end-to-end cnn acceleration. *IEEE Transactions on Computers*, pages 1–1, 2021.