MaxiMals: A Low-cost Hardening Technique for Large Vision Transformers

Lucas Roquet, Fernando Fernandes dos Santos, Paolo Rech, Marcello Traiola, Olivier Sentieys, Angeliki Kritikakou

MaxiMals: A Low-cost Hardening Technique for Large Vision Transformers

Lucas Roquet[†], Fernando Fernandes dos Santos[†], Paolo Rech[‡], Marcello Traiola[†]
Olivier Sentieys[†] and Angeliki Kritikakou[†*]
[†]Univ Rennes, Irisa, INRIA, France
[‡]University of Trento, Italy
[*]Institut Universitaire de France (IUF), France

***Abstract*—We propose MaxiMals, an experimentally tuned low-cost mitigation solution to reduce the impact of transient faults on Vision Transformers. MaxiMals can correct 73.06% of critical failures on average, with overheads as low as 0.25%.**

I. Introduction

Transformers are state-of-the-art ML models that excel in various autonomous system tasks such as language processing, image classification, radar processing, and instance segmentation. Thanks to their ability to learn a wide range of concepts from data, ViTs are especially useful for complex applications like autonomous driving [1] and industrial automation [2]. However, given the complexity of the models and the high number of parameters (which can exceed one trillion [3]), ViTs need to be executed on large hardware accelerators, such as Graphic Processing Units (GPUs). GPUs are the most suitable hardware architecture to train and use large ViT models due to their flexibility and high-performance computing capabilities. GPU vendors have significantly improved their products' computing power, frameworks, and hardware reliability. Modern GPUs feature a tailored Single Error Correction Double Error Detection (SECDED) Error Correction Codes (ECC) in the main memories [4]. Despite ECC, GPUs executing ViTs can still present high neutron-induced fault rates due to their extensive computing resources. The probability of multiple parallel units being simultaneously affected compromises the reliability of ViT-based systems, posing a threat to ViT-based autonomous safety-critical applications.

Unfortunately, traditional mitigation strategies, such as modular redundancy [5] and Algorithm-Based Fault Tolerance (ABFT) [6], become nearly impractical when adapted to large Transformers, since there are billions (even trillions) of parameters and gigabytes of memory to manage and protect. Even a simple float value restriction, applied across all layers of a ViT model, imposes a significant 68.61% overhead [7]. Thus, novel and effective solutions are required to enhance ViT's reliability, such as the one we propose in this paper.

We propose a protection strategy tailored to ViT, targeting only faults more likely to affect the model's accuracy. To identify the faults to correct, we expose an NVIDIA GPU to a neutron beam with a fluence of 4.1×10^{11}, measuring the error rate of 5 large ViT models and characterizing the fault model of ViT basic kernels. Based on the ViT kernels evaluation, we propose a low-cost and effective fault-tolerant mechanism, named *MaxiMum corrupted values* (MaxiMals), that corrects only the critical corrupted floating-point values during the inference. Notably, MaxiMals incurs low execution time overheads, as low as 0.25% (3.53% on average), and requires minimal model modifications while reducing up to 100% of misclassifications (73.06% on average). Specifically, our contributions include:

- A reliability evaluation of 5 ViTs on NVIDIA Pascal architecture (Quadro P2000) using a neutron beam.
- The ViT neutron-induced fault model characterization and how transient faults affect ViT kernels.
- An efficient hardening technique with minimal model modifications and low execution time overhead. The proposed approach, MaxiMals, was validated with neutron beam experiments.

II. Background and Contributions

The ViT model, introduced by Dosovitskiy *et al.* [8], improves image classification accuracy by treating input images as sequences of patches. These patches are transformed using linear transformations before being fed into the model. Additionally, the ViTs have similar structures across their variants like EVA2 [1], SwinV2 [9], and MaxViT [10]. Transformers use Encoder Blocks, including Multi-Layer Perceptrons (MLP), Identity and Normalization layers, and Multi-Head Attention (MHA) networks. While Normalization, Identity, and MLP are conventional kernels of ML, the MHA module, the innovation of Transformers, enables attention to image areas for context understanding. We evaluate the impact of neutron-induced faults (error rate and model) on each ViT kernel to enable efficient fault tolerance for ViTs. Our analysis in Section V shows that ViT's error rate is dependent on memory and computational resources.

Hardware accelerators for ViT on autonomous systems are susceptible to soft errors caused by faults induced by ionizing particles, such as high-energy neutrons [11]. These errors may not damage the device physically but can significantly impact the output of ViT models, potentially changing their final classifications. When not **masked**, soft errors can propagate to the software level and cause **Detected Unrecoverable Errors (DUEs)**, hang the program or crash the entire system, and **Silent Data Corruptions (SDCs)**, that allow the application to complete its execution but with an incorrect output, and without a fault-tolerance method, the failure remains undetected.

Particularly concerning ViT models, SDCs can be further categorized into **Tolerable SDCs**, which modify the model output but not the classification outcome, or **Critical SDCs**, which causes the model to change the top 1 classification probability, resulting in misclassification. We focus on correcting Critical SDCs only to provide efficient mitigation solutions.

In order to provide an efficient hardening, we adopt a strategy that consists of observing, with beam experiments, how (and how often) the hardware fault propagates to a software visible state. We built specific microbenchmarks to evaluate the impact of the transient faults on the ViTs' main operations, MLP, Attention, and Block.

ABFT [6] and value restriction [12]–[14] are established approaches to prevent Critical SDC on ML models. Interestingly, conventional strategies for large Transformers lead to high overheads due to their resource-demanding nature. Researchers have adapted ABFT [15] and range restriction [7] for ViTs. However, a simple range restriction approach for all the layers of a ViT model can add up to 68.61% overhead on execution time [7]. To address this issue, our proposed MaxiMals approach is an experimentally tuned method at the application level that increases fault tolerance for large ViT models with low overhead. This is achieved by targeting only critical faults. We refrain from suggesting hardware design changes, resulting in costly hardware modifications that could affect performance and design time. Instead, we efficiently manage hardware faults at the application level.

III. Experimental Methodologies

In this section, we explain our experimental methodology and the error rate metrics used to evaluate ViT's failure rate and failure criticality.

System Under Test: We performed beam experiments with NVIDIA GPU Pascal architectures (Quadro P2000). The Quadro P2000 is built with TSMC $16nm$ FinFET, featuring an L1 cache of 48KB per Streaming Multiprocessor (SM), an L2 cache of 1280 KB, and 1024 CUDA cores. The GPU has 256 KB registers per SM and a power consumption of up to 75W. Our beam experiments only focus on GPU core errors (beam spot set to 2cm diameter to avoid affecting onboard DRAM).

We evaluated 5 ViT models from the HuggingFace library (v0.8.19) [16]. The models belong to 4 families: Original ViT [8], EVA2 [1], SwinV2 [9], and MaxViT [10]. The models differ in size and input patches. For the experiments, we used a Python program with PyTorch v2.0.0 to load the ViT and perform inferences on a batch of random images from the ImageNet dataset [17].

Table I displays the essential features of the evaluated models, such as their GPU memory usage, accuracy, execution time, and the minimum and maximum output values of Identity layers utilized for the MaxiMals implementation. To obtain the value ranges for the MaxiMals implementation, we ran the ViT models on the entire Imagenet dataset and recorded the minimum and maximum values forwarded through each Identity layer.

Beam Experiments: We measured the neutron-induced error rate of the ViTs from Table I by exposing the GPUs to a neutron beam. The beam experiments provide the Failure In Time (FIT) – the number of faults expected in $10^9 h$ of operation. FIT is calculated by dividing the number of errors by the neutron fluence and then multiplying by the terrestrial neutron flux ($13n/(cm^2 \times h)$) and by 10^9. The experiments were done at the ChipIR facility of the Rutherford Appleton Laboratory, UK. Figure 1 shows the installed setup, consisting of GPUs aligned with the neutron beam and connected to the motherboard. The beam setup utilizes Python scripts to monitor and execute ViT models on a server outside the beam room, while the software is designed to recover from device hangs and restart the program if it fails to respond within a set timeframe. The same ViT model is run on the GPU for several iterations, and any differences between the outputs and a previously saved output are recorded as Tolerable SDC or Critical SDC. The codes used on the beam experiments are disclosed[1].

TABLE I: ViT models size, accuracy, execution times for Pascal GPU, and Imagenet dataset profiled value ranges.

	Patch Size	Size (MB)	Acc. (%)	Time (ms)	Value Range Min	Value Range Max
ViT-L [8]	L14-224	1164	87.90	488	-231.3	124.6
ViT-H [8]	H14-224	2479	88.20	1644	-83.4	90.9
EVA2 [1]	L14-448	1176	89.95	2686	-342.6	327.6
SwinV2 [9]	L-256	787	86.94	404	-22.5	22.7
MaxViT [10]	L-384	845	87.98	938	-66806.8	35259.4

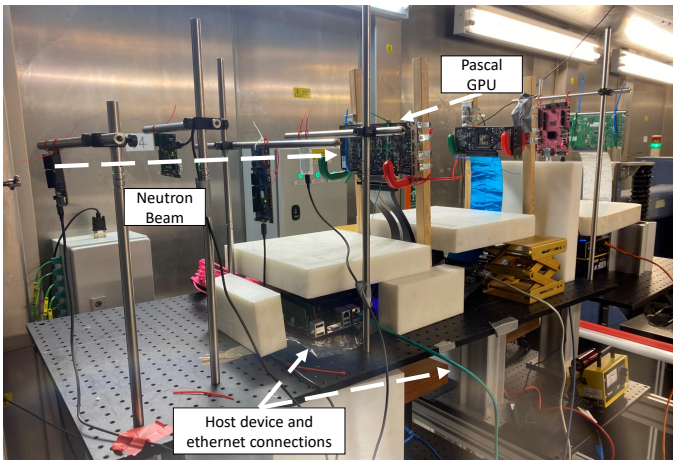

Fig. 1: Quadro P2000 GPUs on ChipIR beamline.

IV. MaxiMals

In this section, we assess the effects of faults induced by neutrons on the main kernels of ViT models (MLP, Attention, and the Transformers Encoder Block). We first demonstrate how errors in ViT operations alter values, then we show how we can prevent these values from generating critical SDCs.

A. ViT' Fault Models

We analyze the reliability of the most common ViT kernels (MLP, Attention, and the Block) to unveil the leading causes of Critical SDCs. Table II shows data from beam experiments

[1]https://github.com/diehardnet/maximals

TABLE II: Experimental data of ViT kernels in the neutron beam experiments for Pascal GPU.

	FIT Rate		*Inf/NaN* (%)	Max value difference
	SDC	DUE		
MLP	22.2 ± 6.7	24.8 ± 7.1	0.0%	1.3×10^{34}
Attention	13.9 ± 3.2	26.2 ± 4.4	10.5%	6.0×10^{34}
Block	25.2 ± 3.6	39.14 ± 4.5	2.9%	1.0×10^{37}

for the kernels extracted from the ViT-L model on a single inference on Pascal GPU. We compute SDC and DUE FIT rates, the percentage of *Not a Number* (*NaN*) and $\pm$*infinity* (*inf*) values observed in all the experiments, and the maximum difference between fault-free and corrupted outputs, for each kernel.

The SDC FIT rates for kernels (on average, 20.49) are higher than the FIT rate of most ViT models (see Section V). This is not surprising, since an SDC in a kernel of the ViT still needs to propagate through the ViT model, and it can still be masked in the downstream Encoder Blocks. That is, we are not yet considering the propagation probability of these faults in the ViT model. Similarly, the DUE FIT rate is higher than the ViT model's (30.08 on average).

The MLP kernel produces no *inf*/*NaN* values. The MLP algorithm is a sequence of multiplying and accumulating instructions and being a simpler algorithm, MLP has less chance of generating *inf*/*NaN* values. Contrarily, the Attention kernel is composed of softmax and division operations. Those operations demand many cycles to compute and are more prone to yield *inf*/*NaN* values, leading to the highest percentage of corrupted values. Lastly, the ViT Block reveals much lower *inf*/*NaN* percentages in the output than the Attention kernel. Attention produces many *inf*/*NaN* values, but, due to masking, these values may not propagate through to the final output of the ViT Block. If corrupted values reach the output of the Block, they can potentially affect the classification of the entire ViT model.

B. Proposed Hardening Approach

The *inf*, *NaN*, and large values pose a risk to ViT's reliability, recent research shows that those corrupted values can reduce the accuracy of a ML model to random guessing [7], [13], [14]. We modified the Identity layers within the ViT Block to prevent the propagation of corrupted values that can generate Critical SDCs. Figure 2a shows a standard ViT Encoder Block, while Figure 2b shows the modified model structures needed to implement MaxiMals.

Using simple object-oriented programming techniques, the MaxiMals approach can be easily implemented for any ViT structure. We create a child class (*HardenedIdentity*) that extends the default Identity layer class. Replacing the default Identity object with the extended HardenedIdentity allows us to effortlessly harden 5 different models described in Table I without any compatibility problem. Then, we execute all the ViTs on the entire ImageNet validation dataset, store the minimum and maximum output values on the Identity layers, and use them as bounds to filter corrupted values. To avoid changing values that are lower/higher false positives, we multiply the profile values by 1.3. If the corrupted value

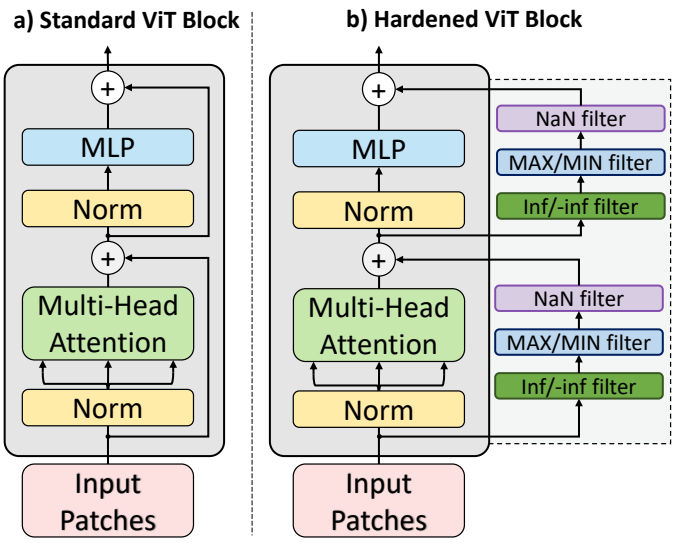

Fig. 2: Unhardened and Hardened ViT Blocks.

TABLE III: Overheads introduced by MaxiMals on ViTs.

	Execution Time Overhead	Added Instructions
ViT-L	3.52%	1.52%
ViT-H	1.72%	1.23%
EVA2	1.37%	0.43%
SwinV2	0.25%	0.07%
MaxViT	10.78%	2.76%

is detected, it is replaced by the lowest or highest value in the case of $\pm$*inf*, and 0 in the case of *NaN*. MaxiMals can be applied to any of the 120,000 models available on the HuggingFace library that uses the default PyTorch modules.

Identity layers neither perform any arithmetic operations nor have learnable parameters. Thus, the performance impact of the MaxiMals is proportional to the number of ViT Identity layers. Table III shows the execution time and additional instructions overheads added by MaxiMals. Our method has a very low overhead in terms of execution time, on average, 3.53%. The worst case is the MaxViT models, which have 384 Identity layers, with a time overhead of 10.78%. We also use NVIDIA profiling tools (Nvprof) to measure the GPU-executed instructions for each ViT model for more precise measurements. MaxiMals increases by up to 2.76% the number of executed instructions.

V. Experimental Validation

In this section, we analyze the FIT rate of ViT and the efficiency of our hardening approach in reducing the critical SDCs rate (i.e., misclassification rate).

Figure 3 shows the experimentally measured SDC (Overall and Critical) and DUE FIT rates for the tested Baseline ViTs models, and the models protected by MaxiMals Values are reported with 95% confidence intervals considering a Poisson distribution.

All ViTs exhibited high DUE FIT rates, on average, the baseline models have DUE FIT rate of 17.60, and the hardened models 14.98. We investigated the causes of DUEs and discovered that memory faults, such as incorrect memory

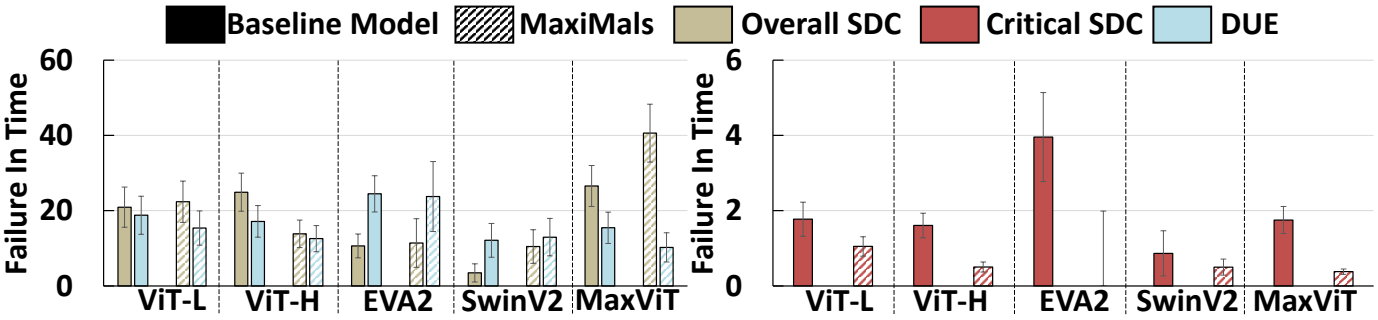

Fig. 3: Overall SDC, Critical SDC, and DUE Failure In Time for the evaluated ViT models (Baseline and MaxiMals).

address accesses and unaligned memory operations, are the source of 84.23% of the DUEs. Due to the resource-demanding nature of Transformers models, their implementation leads to an increased number of global memory accesses and warp scheduling stress, which translates to a high DUE rate.

The SDC FIT rates, as illustrated in Figure 3, are directly affected by the complexity and type of the ViT model. The average Overall SDC FIT rate (Tolerable + Critical SDCs) for baseline models is 17.30, while for MaxiMals-protected ViTs, it is 19.74. Notably, the highest Overall SDC FIT rate is from MaxViT protected by MaxiMals, which is 40.60 FIT. This is not surprising since MaxViT has the highest number of Identity layers of all the models (384 identity layers). Even after correcting any errors by clipping them to the maximum values, the last classification head's output may still differ from the fault-free output. This is because the corrupted values corrected by MaxiMals will differ from the expected values but will not lead to any Critical SDCs. It is important to note that we consider any difference in the classification head's output from the fault-free execution as an SDC.

Critical SDCs' probability depends on various ViT characteristics, such as weight values, accuracy, and activation layers. Despite the high accuracy and significant data redundancy of ViTs, Critical SDCs can still occur, as shown in Figure 3. This is especially evident in the case of large models like EVA2 and ViT-L, which exhibit a Critical SDC FIT rate of 3.95 and 1.77 for the baseline unprotected models, respectively. Models with many identity connections and linear operations, like EVA2, can have a higher criticality than other models due to the ease with which errors can propagate between the layers. On the other hand, the SwinV2 model has the lowest Critical SDC FIT rate, i.e., 0.87 FIT. SwinV2's patches are organized using a "shifted window" that slides through the input, creating overlapped patches, which add more redundancies to the represented data, leading to a more reliable model. Although the Critical FIT rate is low for some models like SwinV2, Critical SDCs can still be extremely dangerous in safety-critical applications.

Figure 3 also shows the effectiveness of the MaxiMals approach. As per the evaluation of various ViT models, the Critical SDC FIT rate of the unprotected versions is always higher than the ones protected by MaxiMals. On average, the Critical SDC FIT rate of unprotected models is 1.99 and the ones protected by MaxiMals have an average of 0.49.

Finally, Table IV presents the percentage of the Critical SDCs for each configuration tested. The proposed approach drastically reduced the Critical SDC percentages, with minimum observed values of 0.93% for MaxViT. This efficiency of MaxiMals on MaxViT is attributed to the number of Identity layers (384 Identity layers over 1097 layers), allowing the filtering of corrupted values at a higher frequency. During the beam experiment campaign, no Critical SDC was observed for EVA2. Consequently, the maximum error bar for the EVA2 Critical SDC was calculated based on Quinn and Tompkins' approach for zero failures [18].

TABLE IV: Critical SDC percentage for the Base ViTs and ViTs hardened with MaxiMals.

	Critical SDC %		Baseline/MaxiMals
	Base Model	MaxiMals	
ViT L	8.47%	4.69%	1.81×
ViT H	6.45%	3.64%	1.77×
EVA2	37.21%	0.00%	3.24×*
SwinV2	25.00%	4.76%	5.25×
MaxViT	6.59%	0.93%	7.05×

*For the MaxiMals value, we consider the maximum error bar value

VI. Conclusions

We conducted a thorough analysis to understand how transient errors induced by neutrons can impact ViT models. Although ViT models are known for their high accuracy, they are resource-intensive and have high SDC and DUE rates. Our evaluation of the most basic operations of ViT has shown that the effects of transient faults on MLP, Attention, and Block can compromise the computation, leading to large values, *NaN/Inf*, and significantly reduce the model accuracy. To address this issue, we have developed a fault tolerance approach called MaxiMals, designed explicitly for ViT models. Our approach reduces Critical SDCs and improves fault tolerance for complex ViT models.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 899546 with the support of the Brittany Region and under the RADNEXT grant agreement No 101008126 [19]. This project is partially funded by ANR FASY (ANR-21-CE25-0008-01) and ANR RE-TRUSTING (ANR-21-CE24-0015-02). ChipIR and RADNEXT provided and supported neutron beam time

experiments (DOI https://doi.org/10.5286/ISIS.E.RB2300036). We acknowledge the researchers Dr. Christopher Frost, Dr. Carlo Cazzaniga, and Dr. Maria Kastriotou, who helped with neutron experiments.

REFERENCES

[1] Y. Fang, Q. Sun, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva-02: A visual representation for neon genesis," *arxiv*, 2023.

[2] C. Chen, C. Liu, T. Wang, A. Zhang, W. Wu, and L. Cheng, "Compound fault diagnosis for industrial robots based on dual-transformer networks," *Journal of Manufacturing Systems*, vol. 66, pp. 163–178, 2023, ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2022.12.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612522002254.

[3] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *Journal of ML Research*, vol. 23, no. 120, pp. 1–39, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-0998.html.

[4] M. B. Sullivan, N. Saxena, M. O'Connor, *et al.*, "Characterizing And Mitigating Soft Errors in GPU DRAM," in *IEEE Micro*, IEEE Micro, 2021, pp. 641–653, ISBN: 9781450385572. DOI: 10.1145/3466752.3480111. [Online]. Available: https://doi.org/10.1145/3466752.3480111.

[5] I. Baek, W. Chen, Z. Zhu, S. Samii, and R. Rajkumar, "FT-DeepNets: Fault-Tolerant Convolutional Neural Networks With Kernel-Based Duplication," in *IEEE WACV*, IEEE WACV, Jan. 2022.

[6] S. K. S. Hari, M. Sullivan, T. Tsai, and S. W. Keckler, "Making convolutions resilient via algorithm-based error detection techniques," *IEEE TDSC*, 2021. DOI: 10.1109/TDSC.2021.3063083.

[7] G. Gavarini, A. Ruospo, S. Ernesto, *et al.*, "Evaluation and mitigation of faults affecting swin transformers," IEEE IOLTS, 2023.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 9th ICLR 2021, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy.

[9] Z. Liu, H. Hu, Y. Lin, *et al.*, "Swin transformer v2: Scaling up capacity and resolution," IEEE CVPR, 2022, pp. 12 009–12 019.

[10] Z. Tu, H. Talebi, H. Zhang, *et al.*, "MaxViT: Multi-axis vision transformer," ECCV, 2022, pp. 459–479.

[11] N. Mahatme, T. Jagannathan, L. Massengill, B. Bhuva, S.-J. Wen, and R. Wong, "Comparison of Combinational and Sequential Error Rates for a Deep Submicron Process," *IEEE TNS*, vol. 58, no. 6, pp. 2719–2725, 2011.

[12] L.-H. Hoang, M. A. Hanif, and M. Shafique, "FT-ClipAct: Resilience Analysis of Deep Neural Networks and Improving Their Fault Tolerance Using Clipped Activation," in *DATE*, DATE, 2020, pp. 1241–1246, ISBN: 9783981926347.

[13] Z. Chen, G. Li, and K. Pattabiraman, "A Low-cost Fault Corrector for Deep Neural Networks through Range Restriction," IEEE/IFIP DSN, Jun. 2021. DOI: 10.1109/DSN48987.2021.00018. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/DSN48987.2021.00018.

[14] N. Cavagnero, F. D. Santos, M. Ciccone, G. Averta, T. Tommasi, and P. Rech, "Transient-fault-aware design and training to enhance DNNs reliability with zero-overhead," in *IOLTS*, IEEE IOLTS, 2022.

[15] K. Ma, C. Amarnath, and A. Chatterjee, "Error Resilient Transformers: A Novel Soft Error Vulnerability Guided Approach to Error Checking and Suppression," in *IEEE ETS*, IEEE ETS, 2023. DOI: 10.1109/ETS56758.2023.10174239.

[16] R. Wightman, *Huggingface*, huggingface.co/timm. DOI: 10.5281/zenodo.4414861.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE CVPR*, IEEE CVPR, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[18] H. Quinn and G. Tompkins, "Measuring zero: Neutron testing of modern digital electronics," *IEEE TNS*, pp. 1–1, 2024. DOI: 10.1109/TNS.2024.3359572.

[19] R. G. Alía, A. Coronetti, K. Bilko, *et al.*, "Heavy ion energy deposition and see intercomparison within the radnext irradiation facility network," *IEEE Transactions on Nuclear Science*, vol. 70, no. 8, pp. 1596–1605, 2023. DOI: 10.1109/TNS.2023.3260309.