Cross-Layer Reliability Evaluation and Efficient Hardening of Large Vision Transformers Models

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

Cross-Layer Reliability Evaluation and Efficient Hardening of Large Vision Transformers Models

Lucas Roquet
Univ Rennes, CNRS, Inria, IRISA
UMR 6074, F-35000 Rennes
France

Fernando Fernandes dos Santos
Univ Rennes, CNRS, Inria, IRISA
UMR 6074, F-35000 Rennes
France

Paolo Rech
University of Trento
Italy

Marcello Traiola
Univ Rennes, CNRS, Inria, IRISA
UMR 6074, F-35000 Rennes
France

Olivier Sentieys
Univ Rennes, CNRS, Inria, IRISA
UMR 6074, F-35000 Rennes
France

Angeliki Kritikakou
Univ Rennes, CNRS, Inria, IRISA
UMR 6074, F-35000 Rennes
Institut Universitaire de France (IUF)
France

Abstract

Vision Transformers (ViTs) are highly accurate Machine Learning (ML) models. However, their large size and complexity increase the expected error rate due to hardware faults. Measuring the error rate of large ViT models is challenging, as conventional microarchitectural fault simulations can take years to produce statistically significant data. This paper proposes a two-level evaluation based on data collected through more than 70 hours of neutron beam experiments and more than 600 hours of software fault simulation. We consider 12 ViT models executed in 2 NVIDIA GPU architectures. We first characterize the fault model in ViT's kernels to identify the faults more likely to propagate to the output. We then design dedicated procedures efficiently integrated into the ViT to locate and correct these faults. We propose *MaxiMum corrupted values* (MaxiMals), an experimentally tuned low-cost mitigation solution to reduce the impact of transient faults on ViTs. We demonstrate that MaxiMals can correct 90.7% of critical failures, with execution time overheads as low as 5.61%.

1 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. When used in vision applications, ViTs selectively focus on essential features in a given frame using attention mechanisms instead of treating all features equally. 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 1 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 GPU memories [4]. Despite ECC in main memories, as we show in this paper, GPUs executing ViTs can still present high neutron-induced fault rates due to their extensive computing resources. Additionally, 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.

Assessing the reliability of ViTs is exceptionally challenging due to the complexity of both the hardware and software framework. While radiation experiments are used as a realistic error rate estimation source [5], they do not allow for tracking fault propagation. Contrarily, with fault simulation, we can pinpoint specific fault sites [6]–[8]. However, fault models are synthetic and provided by the user. It is thus essential to ensure that simulation fault models are realistic to avoid drawing misleading conclusions or implementing ineffective hardening solutions. Recent works show that the results obtained from different fault simulation levels, such as microarchitectural and software, can vary up to one order of magnitude [6], if the fault model does not realistically represent real-world errors. In this paper, we adopt a cross-layer analysis (*novel for ViTs*), combining radiation and software fault simulation approaches, to obtain realistic and traceable insights into ViT reliability.

Unfortunately, traditional mitigation strategies, such as modular redundancy [9] and Algorithm-Based Fault Tolerance (ABFT) [5], 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 [8]. Thus, novel and effective solutions are required to enhance ViT's reliability, such as the one we propose in this paper.

We propose protecting against faults most likely to affect ViT's accuracy. To identify the faults to correct, we expose two NVIDIA GPU architectures to neutron beams for over 70 hours (equivalent to 700,000 *years* of terrestrial operation), measuring the error rate of 6 large ViT models (with and without ECC) and characterizing the fault model of ViT kernels. Then, we conduct over 600 hours of software fault simulation using the NVIDIA Fault Injector [10] tuned with the experimentally observed fault models. We track fault propagation and identify the faults that are more likely to corrupt the ViT inference. Based on our cross-layer analysis, we propose a low-cost and effective fault-tolerant mechanism, named *Maximum corrupted Malicious values* (MaxiMals), that corrects only the critical corrupted floating-point values in the inference. Notably, MaxiMals incurs a low average overhead of 5.61% in execution time and requires minimal model modifications while reducing up to 90.7% of misclassifications (61.41% on average).

Specifically, our contributions include:

- A reliability evaluation of 6 ViTs on 2 GPU architectures using a neutron beam, discussing the dependence of ViT error rates on complexity and trade-off of enabling ECC.
- The ViT neutron-induced fault model characterization and how transient faults affect ViT kernels.
- A detailed analysis of the fault propagation in ViT models based on fault simulation to unveil which faults affect ViT's ability to classify images correctly.
- An efficient hardening technique with minimal model modifications and low execution time overhead.

2 Background and Contributions

The ViT model, introduced by Dosovitskiy *et al.* [11], 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 [12], and MaxViT [13]. 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 4 shows that ViT's error rate is linearly dependent on memory and computational resources.

Hardware accelerators for ViT employed on autonomous systems are susceptible to soft errors caused by faults induced by ionizing particles, such as high-energy neutrons [14]. 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)** or **Silent Data Corruptions (SDCs)**. DUEs hang the program or crash the entire system, while SDCs 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 light of the complexity of ViT models and their high memory and computational requirements, we have opted for reliability evaluation methods that generate accurate data in a feasible amount of time: physical injection with a neutron beam and instruction-level fault simulation. Although lower levels of fault simulation, such as RTL and microarchitectural, can produce more precise outcomes [6], recent studies indicate that characterizing all the requisite fault simulations for large ML models could take months to years [15]. For example, simulating a tiny 5-layer neural network that is around 5.10^3 times smaller than the smallest model we assessed (ViT BS32-224) takes approximately 1.2 hours in a GPU microarchitectural simulator [15]. It would require ≈7 centuries to simulate 1,000 faults on a GPU microarchitectural simulator for ViT BS32-224.

In order to provide a comprehensive evaluation, we adopt a cross-layer strategy that consists of observing, with beam experiments, how (and how often) the hardware fault propagates to a software visible state. Then, we propagate this effect in software to track how it modifies the ViT output. Combining data from neutron beam experiments with instruction-level simulations therefore provides valuable insights into the reliability of large ViT models, and how to enhance their fault tolerance at the application level.

ABFT [5] and value restriction [7], [16] 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 [17] and range restriction [8] 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 [8]. 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.

3 Evaluation Methodology

This section describes our experimental methodology, error rate metrics, and reveals how ViT features affect reliability.

System Under Test: We performed fault simulations and beam experiments on two NVIDIA GPU architectures, Pascal (Quadro P2000) and Ampere (RTX A2000). The Quadro P2000 is built with TSMC 16*nm* FinFET, featuring an L1 cache of 48KB per Streaming Multiprocessor (SM), an L2 cache of 1280 KB, and 1024 CUDA cores. The RTX A2000 is built with TSMC 7*nm* FinFET, featuring an L1 cache of 128 KB per SM, an L2 cache of 3MB, and 3328 CUDA cores. Both GPUs have 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). The RTX A2000 has SECDED ECC to protect the register file and cache memories. Tests were conducted with ECC ON for some ViTs to assess ECC efficacy.

We evaluated 12 ViT models from the HuggingFace library (v0.8.19) [18]. The models belong to 4 families: Original ViT [11], EVA2 [1], SwinV2 [12], and MaxViT [13]. 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 [19]. Table 1 shows essential features of the evaluated models, including their GPU memory usage, accuracy, execution time, and the min and max Identity layers' output values used for the MaxiMals implementation.

Physical Fault Injection: We measured the neutron-induced error rate on a subset of ViTs from Table 1 by exposing the GPUs to a neutron beam. As we could not assess all 12 configurations due to beam time limitations, *the original ViT configurations were prioritized*. The beam experiments provide the Failure In Time (FIT), which is calculated by dividing the number of errors by the neutron fluence and then multiplying by the terrestrial neutron flux (13×10^9). 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

Table 1. ViTs' memory size, accuracy, execution times for Pascal (P.) and Ampere (A.) GPUs and profiled values.

		Size	Acc.	Time (ms)		Value Range	
		(MB)	(%)	P.	A.	min	max
ViT [11]	BS32-224	339	73.6	31	7	-32.8	37.6
	B16-224	333	84.5	135	24	-35.1	63.5
	B16-384	340	85.4	461	72	-55.6	67.0
	L14-224	1164	87.9	488	106	-231.3	124.6
	H14-224	2479	88.2	1644	216	-83.4	90.9
EVA2 [1]	B14-448	350	88.6	812	161	-904.2	483.7
	L14-448	1176	89.9	2686	509	-342.6	327.6
SwinV2 [12]	B-256	372	86.2	182	53	-18.6	19.0
	B-384	431	87.1	579	172	-18.6	18.5
	L-256	787	86.9	404	92	-22.5	22.7
MaxViT [13]	L-384	845	87.9	938	282	-66806.8	35259.4
	L-512	856	88.0	1762	518	-83250.6	40806.9

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].

Software Fault Simulation: We used the NVIDIA Bit Fault Injector (NVBitFI) [10] for the instruction-level fault simulation. NVBitFI allows simulating faults at the Shader Assembly level (SASS), i.e., GPU kernel at the assembly level. NVBitFI allows choosing different fault models and sites to evaluate the ViT models. While injecting random bit flips in software fault injection is not accurate [6], injecting in software an experimentally-tuned fault model (i.e., the observed manifestation of the hardware fault in a software visible state) has been proved accurate for GPUs [20]. We inject faults in general-purpose registers, memory load instructions, and arithmetic floating-point operations. We used different bit fault models derived from beam experiments, detailed in Section 4. We simulate 1,750 faults per ViT model, equivalent to more than 600 hours of simulations. The failures (SDCs and DUEs) are counted similarly to beam experiments. With fault simulations, we measure the Program Vulnerability Factor (PVF) for each ViT model. The PVF is the probability of an injected fault propagating from the assembly instruction to the application output [21].

4 Fault Effects on ViT Models

In this section, we analyze the ViT's FIT rate and identify the causes of Critical SDCs. We show that single-bit flip fault simulations are insufficient for ViT reliability assessment.

ViT's FIT rate: Figure 2 shows the experimentally measured SDC (Tolerable and Critical) and DUE FIT rates for the tested ViTs configurations. Values are reported with 95% confidence intervals considering a Poisson distribution.

All ViTs show high DUE FIT rates, on average, 20.50 for Pascal and 32.57 for Ampere ECC OFF. Investigating the cause for DUEs, we found that, when ECC is OFF, on average, 59.39% of the DUEs for Pascal and 78.14% of the DUEs for Ampere are caused by memory faults, such as incorrect memory address accesses and unaligned

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

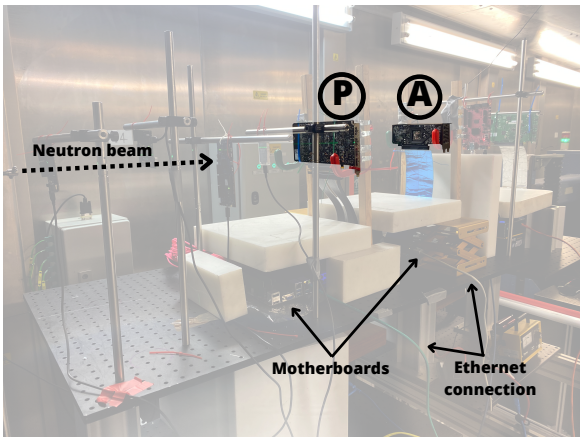


Figure 1. P2000 (**P**) and A2000 (**A**) GPUs on ChipIR beamline.

memory operations. The differences in the DUE rate for Pascal and Ampere are attributed to the lower resources available on Pascal, forcing more global memory accesses and warp scheduling stress. On Ampere GPU, when ECC is ON, the DUE FIT rate increases by an average of 1.84× due to exceptions triggered by uncorrectable double-bit flips detected by ECC. Those faults account, on average, for 77.10% of the total DUEs.

As shown in Figure 2, whereas the DUE FIT rate does not significantly depend on the model, the SDC FIT rates directly depend on the ViT model complexity and the related GPU resource utilization. The average SDC FIT rate for the Pascal GPU is 6.66, with the highest SDC FIT rate being 9.55 (EVA2 B14-448). For Ampere GPU, when ECC is OFF, the average SDC FIT is 30.74, and the highest is 50.62 (H14-224). The trend is less evident for Pascal, for which we observe only a slight increase in the SDC rate. This is justified by resource saturation in the Pascal GPU (even the smaller ViT uses all GPU resources), while in the Ampere GPU, bigger models use more parallel resources, increasing the SDC FIT rate.

The likelihood of Critical SDCs occurring depends on various ViT characteristics like 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 2. This is particularly evident in the case of large models, such as EVA2 B14-448 and ViT L14-224, which exhibit a percentage of Critical SDC of 33.33% and 37.33%, respectively. Models with many residual connections and linear operations, like EVA2, can have higher criticality than other models due to the ease with which errors can propagate between the layers. Interestingly, H14-224 has a low Critical SDC rate when executed with ECC OFF, 7.14% for Pascal and 6.17% for Ampere. This can be attributed to the model having a massive amount of parameters (2.48GB), making it less likely for the corruption of a single or a small group of parameters to result in Critical SDCs.

When ECC is ON, the SDC FIT rates are reduced by 71.51%. Nonetheless, even with ECC ON, the ViT L14-224 still experiences Critical SDCs in 10.0% of the cases. ECC does not provide full protection from Critical SDCs and increases the DUE rate. Thus, alternative hardening solutions are required.

ViT's Fault Models: We analyze the reliability of the most common ViT kernels (MLP, Attention, and the Encoder Block) to unveil the leading causes of Critical SDCs. Table 2 shows data from

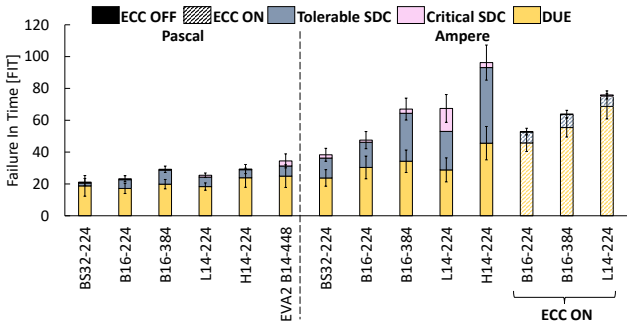


Figure 2. Pascal and Ampere GPUs experimentally measured Tolerable and Critical SDC, and DUE FIT rates for the ViTs.

beam experiments for the kernels extracted from the L14-224 model on a single inference on Pascal GPU. We compute SDC and DUE FIT rates, the percentage of *Not a Number* (*NaN*) and ±*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 modules. 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. Conversely, the DUE FIT rate is comparable to the ViT model's (30.08 on average). This is because a DUE in a kernel will hang the kernel execution and, consequently, the application in which the kernel is being used. In other words, when a crash/hang occurs, it cannot be masked.

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.

Software Fault Simulation: To realistically evaluate which faults - among those observed in beam experiments on the kernels - propagate and generate Critical SDCs, we conducted fault simulation campaigns employing multiple fault models. Moreover, the goal is to identify the models that better represent the neutron-induced faulty effect. For instance, for FP32 instructions, we used a customized version of NVBitFI to simulate a more complex fault model that writes a random value on all the threads on a GPU warp. Multiple threads fault models accurately emulate GPU faults that threaten ML models' reliability, as demonstrated in [20].

Figure 3 shows the average probability for the fault models injected in the 4 ViT families to induce a Critical SDC. We also plot the results for the MaxiMals hardened version (described in Section 6) to introduce the efficacy of our method. We include in Figure 3 the average Critical SDC rate obtained with neutron beam experiments (for Original ViT and EVA2). The similarity of the rates obtained

Table 2. Experimental data of ViT kernels in the neutron beam experiments for Pascal GPU.

	FIT Rate		*Inf/NaN*	Max val.
	SDC	DUE	*(%)*	diff
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}

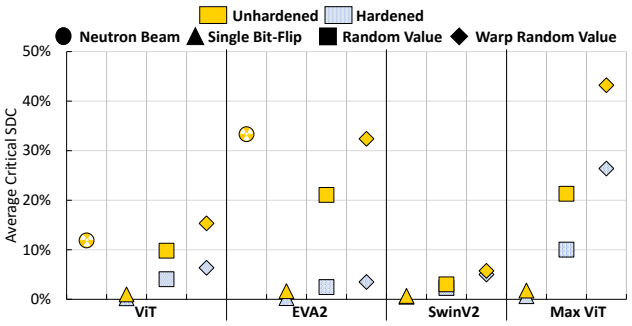


Figure 3. Average Critical SDC per fault model and ViT type.

with beam and fault injection further strengthens the accuracy of our fault simulation. Figure 3 shows that single-bit flips injected at the instruction level do not provide a realistic evaluation for ViT models since most are masked and produce a Critical SDC rate close to zero, well off the rate obtained with beam experiments. Note that only 3% of single-bit flip fault injections resulted in values higher than 10^6 after the fault mask was applied to the target register. In contrast, when random values were used, 45% of the fault mask immediately produced values higher than 10^6. Random and warp random values closely match the beam experiment results. Single particle corruption in shared memory or warp scheduler can cause faults leading to *inf*, *NaN*, and large values. If these faults spread to ViT structures, they may lead to Critical SDCs. Our hardening strategy is designed based on these observations.

5 MaxiMals

The previous sections established that *inf*, *NaN*, and large values pose a risk to ViT's reliability. We modified the Identity layers within the ViT Block to prevent the propagation of corrupted values that can generate Critical SDCs. Fig. 4a shows a standard ViT Encoder Block, while Fig. 4b 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 12 different models described in Table 1 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 is detected, it is replaced by the lowest or highest value in the case of ±*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.

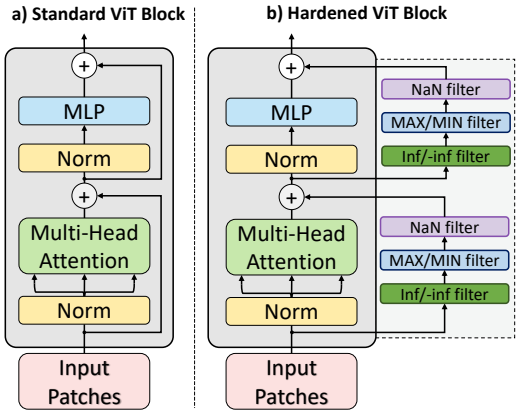


Figure 4. Unhardened and Hardened ViT Blocks.

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 3 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, 5.61%. The worst case is the MaxViT models, which have 384 Identity layers, with a time overhead of 16.09%. We also use NVIDIA profiling tools (Nvprof/Nsight Systems) to measure the GPU-executed instructions for each ViT model for more precise measurements. MaxiMals increases by up to 3.19% the number of executed instructions.

6 Experimental Validation

In this section, we present the validation of MaxiMals. Figure 5 depicts the SDC (Tolerable and Critical) and DUE PVFs for the 12 tested ViTs models, unhardened and hardened versions. The fault simulations are performed both on Pascal (Fig. 5a) and Ampere (Fig. 5b) GPUs.

While the unhardened original ViT models exhibit the highest SDC PVF (on average, 38.97% on Pascal GPU and 34.20% on Ampere GPU), more complex ViTs like unhardened MaxViT effectively mask more faults. MaxViT has the lowest SDC PVF, on average 24.22% on Pascal GPU and 30.51% on Ampere GPU. However, despite the masking ability of more complex models, the Critical SDCs still propagate. For the unhardened models, the MaxViT and EVA2 show the highest Critical SDC PVF, on average, 16.05% for MaxViT and 14.35% for EVA2. Due to their distinct Transformers architecture, these models are especially susceptible to faults such as random and warp random values. MaxViT and EVA2 introduce many improvements, such as different normalization layers (e.g., sub-LN), additional connections for positional information injection, and convolution blocks before Attention layers. These additional modules facilitate the propagation of corrupted values, leading to an increase in Critical SDCs. In contrast, the SwinV2 model manifests the lowest Critical SDC PVF, on average, 2.40%. 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.

MaxiMals reduces the Critical SDC, on average, from 8.50% to 3.28%. For the most complex models, EVA2 and MaxViT, our approach lowers the Critical SDC PVF, on average, to 1.65% and 8.54%, respectively. For EVA2 B14-448, the Critical SDCs are reduced by 90.70% (from 12.91% to 1.20%). The notable success of the MaxiMals technique in models like EVA2 and MaxViT can be attributed to the number of Identity layers (necessary for their complex architecture), allowing the filtering of corrupted values at a higher frequency.

Table 3. Added overheads for ViT model families (min–max)

	Execution Time		Instructions	
	Pascal	Ampere	Pascal	Ampere
ViT	1.7%–3.5%	5.1%–10.0%	1.2%–2.0%	1.0%–1.7%
EVA2	1.3%–5.3%	2.3%–9.4%	0.4%–1.5%	0.4%–1.8%
SwinV2	0.1%–0.5%	0.3%–0.8%	0.1%–0.1%	0.1%–0.1%
MaxViT	9.5%–10.7%	15.3%–16.0%	2.7%–2.7%	3.1%–3.1%

MaxiMals exhibits lower effectiveness in reducing the Critical SDCs for SwinV2. The average Critical SDC for SwinV2 models is reduced by 21.82%. SwinV2 possesses an intriguing characteristic of having few Identity layers compared to other models (only 5), lowering the effectiveness of our approach. This observation aligns with earlier studies that applied value restrictions to SwinV2 [8]. To implement value restriction on SwinV2 models, comprehensive modifications across all layers of the ViT would be necessary, incurring an impractical overhead of 68.61% [8].

Figure 3 illustrates the efficacy of MaxiMals approach in dealing with various fault models that generate Critical SDCs. Our proposed hardening approach is effective in these cases, as it reduces the Critical SDC, on average, of 65.87% for random value and 57.31% for warp random value. Note that our method can be further enhanced for models like SwinV2, where the Identity layers are less frequent. In the SwinV2 scenario, the Blocks can be selectively strengthened by increasing the frequency of value restriction operations based on their criticality without affecting performance.

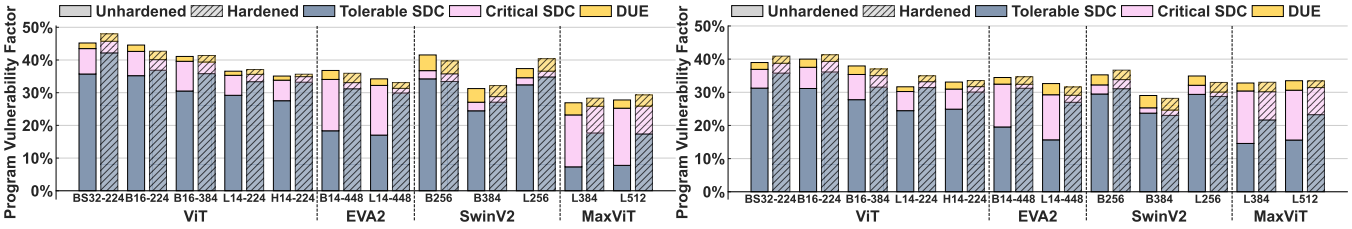
Ultimately, the proposed MaxiMals has kept the DUE PVF on fault simulation the same for the unhardened and hardened models, on average, 2.55% and 2.48%, respectively.

7 Conclusions

We conducted an extensive analysis to understand how transient errors induced by neutrons can affect ViT models, potentially leading to Critical SDCs. Despite ViT's high accuracy, these models are notably resource-intensive and exhibit high SDC and DUE rates. Our comprehensive fault propagation analysis shows the impact of different types of faults on ViT models' classification accuracy. Our findings suggest that single-bit flip faults have minimal impact, while severe faults can significantly degrade accuracy. To address this, we developed a fault tolerance approach called MaxiMals, tailored explicitly to 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 [22]. 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



(a) Pascal GPU **(b)** Ampere GPU

Figure 5. Tolerable SDC, Critical SDC, and DUE Program Vulnerability Factors for the unhardened and hardened ViT models.

Cazzaniga, and Dr. Maria Kastriotou, who helped with neutron experiments.

References

[1] Y. Fang *et al.*, "Eva-02: A visual representation for neon genesis," *arxiv*, 2023.

[2] C. Chen *et al.*, "Compound fault diagnosis for industrial robots based on dual-transformer networks," vol. 66, pp. 163–178, 2023.

[3] W. Fedus *et al.*, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *JML*, vol. 23, no. 1, Jan. 2022.

[4] M. B. Sullivan *et al.*, "Characterizing And Mitigating Soft Errors in GPU DRAM," in *IEEE Micro*. 2021, pp. 641–653.

[5] S. K. S. Hari *et al.*, "Making convolutions resilient via algorithm-based error detection techniques," *IEEE TDSC*, 2021.

[6] G. Papadimitriou *et al.*, "Demystifying the system vulnerability stack: Transient fault effects across the layers," in *IEEE ISCA*, 2021, pp. 902–915.

[7] Z. Chen *et al.*, "A Low-cost Fault Corrector for Deep Neural Networks through Range Restriction," in *IEEE/IFIP DSN*, Jun. 2021.

[8] G. Gavarini *et al.*, "Evaluation and mitigation of faults affecting swin transformers," in *IEEE IOLTS*, 2023.

[9] I. Baek *et al.*, "FT-DeepNets: Fault-Tolerant Convolutional Neural Networks With Kernel-Based Duplication," in *IEEE WACV*, Jan. 2022.

[10] T. Tsai *et al.*, "NVBitFI: Dynamic Fault Injection for GPUs," in *IEEE/IFIP DSN*, 2021, pp. 284–291.

[11] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2021.

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

[13] Z. Tu *et al.*, "MaxViT: Multi-axis vision transformer," in *ECCV*, 2022, pp. 459–479.

[14] N. Mahatme *et al.*, "Comparison of Combinational and Sequential Error Rates for a Deep Submicron Process," *IEEE TNS*, vol. 58, no. 6, pp. 2719–2725, 2011.

[15] J. S. Lew *et al.*, "Analyzing machine learning workloads using a detailed GPU simulator," *arxiv*, 2018.

[16] L.-H. Hoang *et al.*, "FT-ClipAct: Resilience Analysis of Deep Neural Networks and Improving Their Fault Tolerance Using Clipped Activation," in *DATE*, 2020, pp. 1241–1246.

[17] K. Ma *et al.*, "Error Resilient Transformers: A Novel Soft Error Vulnerability Guided Approach to Error Checking and Suppression," in *IEEE ETS*, 2023.

[18] R. Wightman, *Huggingface*, huggingface.co/timm.

[19] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *IEEE CVPR*, 2009, pp. 248–255.

[20] C. Bolchini *et al.*, "Fast and accurate error simulation for cnns against soft errors," *IEEE TC*, vol. 72, no. 4, pp. 984–997, 2023.

[21] V. Sridharan *et al.*, "Eliminating microarchitectural dependency from Architectural Vulnerability," in *IEEE HPCA*, 2009.

[22] R. G. Alía *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.