Assessing the Limitations of Activation Clipping for Fault Mitigation in Vision and Language Transformers

Lucas Roquet, Fernando Fernandes, Luigi Carro, Angeliki Kritikakou

Assessing the Limitations of Activation Clipping for Fault Mitigation in Vision and Language Transformers

Lucas Roquet
IRISA/Inria
Univ Rennes
Rennes, France

Fernando Fernandes dos Santos
IRISA/Inria
Univ Rennes
Rennes, France

Luigi Carro
Institute of Informatics
UFRGS
Porto Alegre, Brazil

Angeliki Kritikakou
IRISA/Inria
Univ Rennes
Rennes, France

***Abstract*—Deep Neural Network (DNN) activation clipping is a well-established method for mitigating hardware-induced faults during inference. Clipping constrains activations to predefined ranges and filtering corrupted values, including *NaN* and *Inf*. However, high variability in activation distributions, both across and within layers of a model, makes it challenging to define fixed ranges that reliably capture corrupted values beyond extreme cases. In this study, we show that activation clipping alone is insufficient to protect vision and language Transformers from inference faults, and that inputs exhibit markedly different levels of fault sensitivity depending on the model's confidence. Our experiments show that the missclassification rate can reach 10.28% even with per-layer clipping is employed.**

I. Introduction

Deep learning achieves unprecedented accuracy across complex tasks, such as computer vision (CV), natural language processing (NLP), and autonomous driving. These advances are driven by the rapid scaling of large Transformer-based DNNs, whose parameter counts can grow from millions to trillions. Following a similar trend, the complexity of the input space is also increased. Modern Transformers operate across diverse modalities and handle highly heterogeneous inputs. Their ability to generalize across complex tasks enables their deployment in mission- and safety-critical applications, where performance, accuracy, and reliability are essential.

Deploying large Transformers relies on powerful hardware accelerators, such as GPUs and TPUs, which comprise thousands of processing cores and many levels of memory hierarchy. This architectural complexity, combined with increasing device density and shrinking transistor sizes, increases susceptibility to hardware faults. Faults induced by radiation, temperature, voltage fluctuations, or material defects can propagate through the computation and disrupt the model's inference [1], [2]. As hardware faults can manifest at the application level as corrupted parameters or activations with extreme outlier values (*inf*, *NaN*, or very large values), a low overhead solution that has been adopted in recent literature is *value clipping* [3]–[9]. By constraining activations or parameters to predefined ranges, value clipping removes fault-induced outliers while preserving DNN nominal behaviour.

However, activation clipping faces key limitations when applied to Transformers. The activations exhibit high variability both across and within layers, making it challenging to define global or per-layer ranges that are simultaneously safe and effective. Tight ranges risk degrading accuracy by clipping correct activations, whereas loose ranges allow corrupted values to propagate undetected. In fact, activation clipping ranges can be defined at different levels, such as model level (a single value range for the whole model), at the tensor or layer level (each tensor or layer will have its value range), or ultimately at activation level (each activation will have its value range) [9]. Although finer-grained strategies can better adapt to activation variability, their computational cost is prohibitive for large-scale Transformers. For example, applying per-activation clipping to a small Transformer (GPT2 Small, 124M parameters) would require storing separate numerical bounds for each activation. Given that GPT2 peak activation is *74,565,312* activations in its Attention heads [10], maintaining per-activation value ranges would impose an impractically large memory overhead.

In this work, we demonstrate that faults that generate values within valid ranges can still lead to misclassification in models protected by activation clipping. We focus on per-layer activation clipping as a case study, as it balances feasibility and activation clipping effectiveness. Our analysis across four Transformer models (GPT2, BART, ViT, and Swin) and two classification tasks (CV and NLP) reveals that activation magnitudes span several orders of magnitude (from -10^3 to 10^3) and exhibit substantial heterogeneity across layers. As Section IV shows, this variability fundamentally limits the protection offered by clipping. Corrupted activations that remain within clipping ranges can propagate undetected, leading to up to 10.28% misclassification even when clipping is applied at every layer. Moreover, the diversity of input samples also affects the effectiveness of clipping. For example, in Vision Transformers (ViTs), factors such as lighting, background, and other image attributes can significantly influence the model's confidence. Even when the `TOP1` class is correct, the predicted probability may be close to other classes, resulting in low-confidence outputs. These low-confidence predictions are particularly vulnerable, as faults that do not produce extreme activation values can still alter the final classification. As shown in Section IV, activation clipping is far more effective for high-confidence inputs and considerably less effective when confidence is low. Overall, the main contributions of

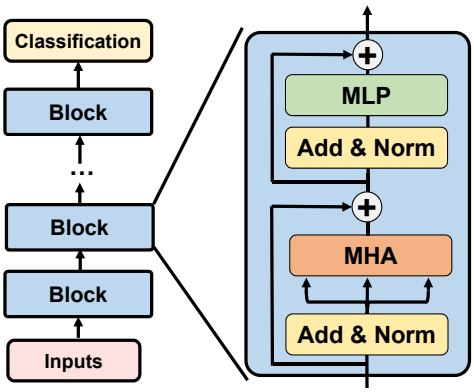

Fig. 1: Typical structure of Transformers with main operations: Block, Multi-Head Attention layers (MHA), Multi-Layer perception (MLP), and Normalisation layers (Add & Norm).

this paper are summarized in the following key insights:

Insight 1: Transformer activation ranges and distributions are highly heterogeneous, both across and within individual layers, limiting activation clipping effectiveness.

Insight 2: Corrupted activations that remain within the clipping ranges can cause misclassification even when per-layer clipping is applied.

Insight 3: The occurrence of missclassification and the effectiveness of activation clipping strongly depend on the model's prediction confidence.

II. DNN Reliability and Hardening

A. Fault Effects on Transformers

Transformers are large DNNs trained on massive datasets that can scale from millions to trillions of parameters [10], [11]. Their core computational unit is the *Transformer Block*, depicted in Figure 1. A model consists of many such blocks stacked sequentially. An input is processed layer by layer until it reaches the final classification head. A standard Block contains three operations: Multi-Head Attention (MHA), a Multi-Layer Perceptron (MLP), and normalization layers. This model structure enables Transformers to achieve high accuracy at the cost of substantial compute and memory demands.

Hardware faults can change the correct execution of Transformers-based systems [12], [13]. Faults occurring in low-level hardware structures, such as registers, pipelines, or scheduler units, can propagate through the computation and manifest as *errors*, including incorrect memory values, invalid branches, or erroneous computation. As errors propagate, they can manifest as *failures*, such as crashes, OS hangs, or incorrect outputs. In Transformer models, an incorrect output on the final classification layer can be classified as a **silent data corruption (SDC)**, and because Transformers' inference is probabilistic (the classification layer produces a ranked list of probabilities for each class, with the `TOP1` value determining the predicted class), not all SDCs lead to misclassification. SDCs can be further classified as **Critical SDCs**, which occur when a fault changes the top predicted class, leading to missclassification, and **Tolerable SDCs**, which modify internal values without altering the final classification. In this work, we focus on evaluating the effectiveness and limitations of activation clipping in preventing *Critical SDCs*. Although Tolerable SDCs occur more frequently [1], Critical SDCs pose the most severe reliability threat in safety-critical systems.

B. Value Clipping for Fault Mitigation

To mitigate the impact of faults in DNNs, several algorithm- or system-level approaches have been proposed, that are based on full or partial redundancy [14], algorithm modification [15], fault-aware training [16], and value clipping [3]–[7], [9]. Compared to other approaches, value clipping is attractive for its low overhead and straightforward integration, which can improve reliability [6] and security [17].

Activation clipping limits the effect of large *outliers* on inference (values far from the majority of activations). A fault can corrupt an activation value represented in FP8, FP16, FP32, or BFLOAT16, turning it into extremely large, *inf*, or *NaN* values. This fault effect transforms a *safe value* into an *unsafe value*. As these corrupted values propagate through the DNN layers, they can corrupt other activations, leading to incorrect predictions. The clipping method assesses whether the activation values remain within the safe ranges. If they exceed these ranges, they are either clipped to fit within the specified range or set to 0. Different value-clipping methods aim for similar goals but differ in granularity and retraining requirements. They can be categorized into two types: *profile-based clipping* and *training-aware clipping*.

1) Profile-based clipping: This class of clipping methods consists of profiling the model with a specific dataset or a set of inputs for calibration and defining a valid value range at a certain granularity, such as the instruction (hardware or software), activation, tensor operation, or layer level. The *Ranger* method adds intermediary layers to clip values outside valid ranges, achieving 97% Critical SDC coverage with only 0.53% overhead [5]. Another example at the layer level is activation clipping based on monitoring minimum, maximum, average, and standard deviation values obtained from dataset profiling [6]. In the context of Transformers, recent works have demonstrated that clipping can be used for fault mitigation [12], [13]. While these methods effectively filter out corrupted values outside the valid ranges, they remain limited in detecting faults whose effects produce values within the accepted valid ranges.

2) Training-aware clipping: One main drawback of profile-based clipping is that it relies on static weights and activation values after training, preventing optimization of value ranges during training. To mitigate this, *training-aware clipping* imposes range constraints during model design and training phases. *FT-ClipAct* [4] replaces ReLU with a bounded version during design, providing similar protection with negligible application-level overhead for CNNs. Similarly, *FitAct* [9] applies per-activation clipping with finetuned thresholds for each activation. While "per-activation" is an effective approach for small neural networks, it becomes prohibitive for large DNNs. *Median Filter* [18] applies a median filter to the fully

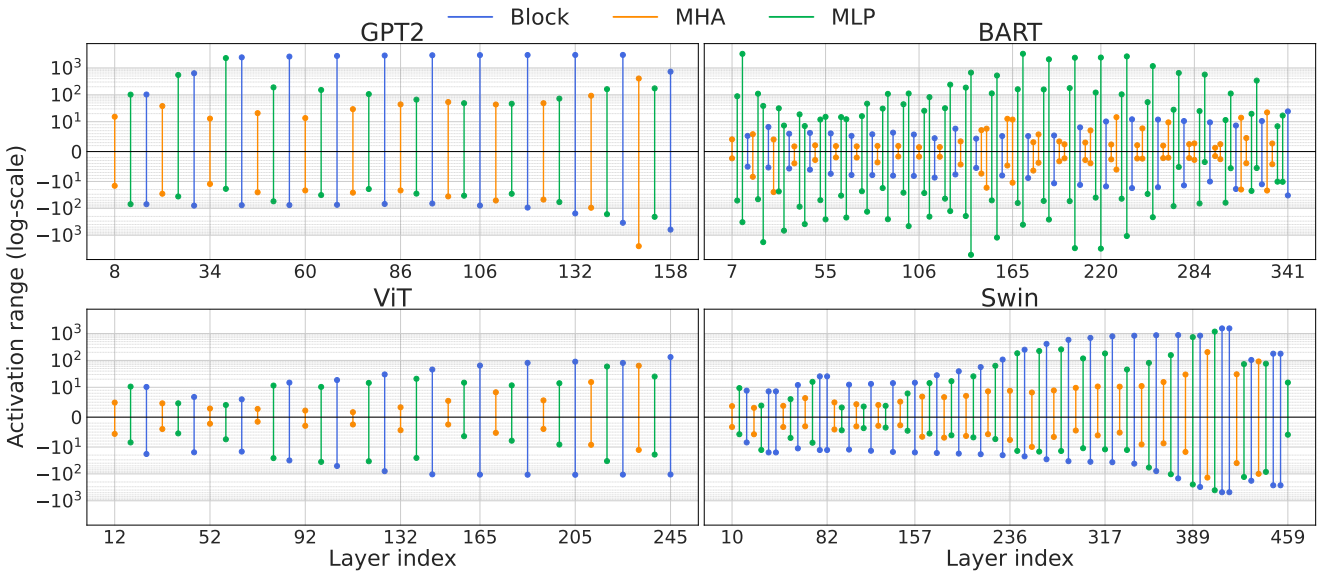

Fig. 2: Activation ranges per core operations (Block, MHA, and MLP) of GPT2, BART, ViT, and Swin.

connected layers and feature maps of CNNs. *Adaptive Clipper* replaces the ReLU activation with a bounded Tanh (Hard-Tanh), requiring additional training steps [19]. *ReLUMAX* [7] enhances ReLU by learning fault-tolerant numerical bounds during training, enabling finetuned clipping of transient faults during inference. While these methods can achieve low overheads (as low as 1%), they are highly dependent on the DNN architecture and task. Clipping-aware retraining and fine-tuning may be non-trivial and can significantly increase training time compared to unprotected versions [9], [20]. This issue is particularly critical for Transformers.

III. Transformers Models Activation Ranges

Transformers achieve high accuracy across diverse tasks (CV and NLP) but produce many *high activation values*. Compared to other DNNs, they exhibit broader activation ranges, a challenge also reported in model quantization [21], [22]. Such extreme values are difficult to constrain with small fixed bounds. In this section, we quantify activation variability for four Transformers on CV and NLP classification tasks.

A. Profiling Activations Setup

Models and datasets: For NLP, we used GPT2-S [10] and BART-L [11], referred to as GPT2 and BART. Both models were fine-tuned on MNLI [23] to classify a sentence as entailment, neutral, or contradiction. For CV, we used two image classification Transformers: ViT-B/16-224 [24] and Swin-B/224 [25], referred as ViT and Swin. Both were evaluated using ImageNet-1K pretrained weights [26]. Table I summarizes the models, and experiments use the MNLI validation set (9,815 inputs) and ImageNet validation set (50,000 inputs).

System configuration: All experiments were executed on Grid'5000 GPU nodes using NVIDIA Ampere GPUs (A40, A5000, A100). We used Python 3.9.2, PyTorch 2.1.1, and pretrained models from HuggingFace.

TABLE I: Models' number of layers, accuracy, number of parameters, number of correctly predicted inputs, and number of inputs in each confidence subset.

	#layers	Acc. (%)	#params. (M)	Correct Pred.	Input subsets			
					Very Low	Low	High	Very High
GPT2	161	81.85	124	8034	463	586	1029	5956
BART	346	90.18	406	8851	142	202	373	8134
ViT	252	84.50	87	42552	2406	3197	6746	30203
Swin	463	85.20	88	42637	2338	3002	6102	31195

B. Activation Value Ranges

We profiled activation values for each layer from each model on their full validation sets. Profiling activation ranges and distributions allows a better understanding of activation variations within the same model and dataset.

Fig. 2 shows per-layer activation ranges for the assessed models. The x-axis is the layer index, and the y-axis is the range of activation values (log-scale). GPT2 shows relatively stable Block activation ranges across layers, while its MHA and MLP layers have substantially different orders of magnitude, with MLP ranges increasing in deeper layers due to amplification effects. BART shows the largest variability, with MLP ranges spanning 10^1 to 10^3, while Block and MHA remain constrained. ViT displays the most stable behavior, with compact activation ranges (within $\pm 10^2$) across all layers, reflecting the strong normalization strategy of ViT. In contrast, Swin activation ranges widen noticeably with depth, especially for Block outputs, mainly due to the cumulative effect of shifted-window MHA.

Activation ranges can demonstrate substantial variations within the same operation. It is the case for MLP layers in GPT2 and BART, which differ by orders of magnitude for the same model. For CV models, the Block shows the highest variation, with ranges increasing with model depth. This intra-operation variability means that layers performing the same operation can operate in very different ranges, making it

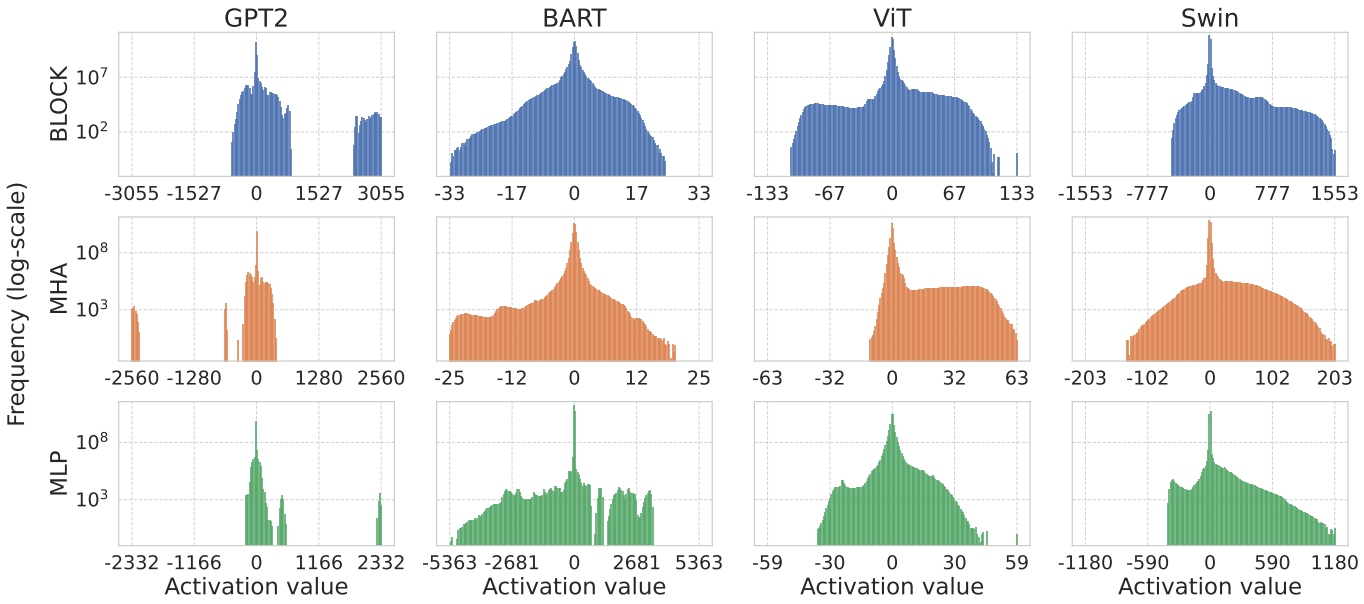

Fig. 3: Activation distributions for the three main Transformers operations (Block, MHA, MLP) across the four evaluated models. Distributions are computed over the full MNLI (GPT2, BART) and ImageNet (ViT, Swin) validation sets.

difficult for a single per-operation clipping range to remain both tight and effective across all layers.

Fig. 3 shows the distributions of activation values for the three main Transformer operations (Block, MHA, and MLP) for GPT2, BART, ViT, and Swin. Each subplot reports the observed activation ranges on the x-axis and the log-scale frequency on the y-axis.

The Block activation ranges are $[-620, 3055]$ for GPT2, $[-33, 24]$ for BART, $[-108, 133]$ for ViT, and $[-481, 1553]$ for Swin. Similar variability is observed in the MHA and MLP layers. Within each model, operations differ strongly. GPT2 displays discontinuous distributions, with dense mass near zero and sparse extreme activations. BART shows narrow activation ranges for Block ($[-33, 24]$) and MHA ($[-25, 22]$) layers but wide MLP ranges ($[-5363, 3370]$). ViT maintains consistent and compact ranges across operations ($[-108, 133]$ for Block, $[-13, 63]$ for MHA, and $[-36, 59]$ for MLP), whereas Swin exhibits significantly larger magnitudes, especially in deeper layers ($[-481, 1553]$ for Block, $[-137, 203]$ for MHA, and $[-406, 1080]$ for MLP).

Insight 1: Activation ranges are highly heterogeneous

Activation ranges in Transformers vary substantially across and within layers. This makes it challenging to define "safe" or "unsafe" values. Global clipping ranges filter only extreme outliers, such as *inf* or large values, leaving faulty activations undetected. Per-layer ranges are still too coarse because layer activation ranges are very wide.

IV. Clipping Limitations: Empirical Analysis

In this section, we present the limitations of activation clipping as a fault-tolerance mechanism for Transformer models.

A. Reliability Assessment Setup

We perform software fault injection (SWFI) to assess the reliability of the Transformers. All SWFI experiments were performed using only the correctly predicted inputs of the validation datasets (Table I). In total, we injected 1,807,650 faults on GPT2, 2,655,300 faults on BART, 9,574,200 faults on ViT, and 9,593,325 faults on Swin. To perform this large-scale SWFI, we used 8 nodes equipped with NVIDIA Ampere GPUs, totaling more than 900 GPU-hours. Faults are injected into the three core Transformer operations (Block, MHA, MLP). Because Transformers contain a large number of layers (161 – 463 layers per model), for illustration purposes, we inject faults at the first, middle, and last layer. Note that, the goal of the SWFI campaign is to identify counterexample faults for which activation clipping fails to suppress the propagation of corrupted activation values, leading to misclassifications.

While SWFI enables efficient fault simulation at a large scale, simple fault models, such as single-bit flips, are known to be simplistic and may not reflect realistic hardware fault behavior [27]. Therefore, we adopt fault models that recent studies have shown to be representative of how low-level hardware faults propagate through activation computations in large accelerators such as GPUs [28]–[30]. We consider three fault models: i) a single activation value is corrupted and propagates across layers, corrupting a large portion of activation values [28]; ii) fault model where a row or column of the layer activation tensor is corrupted [29]; iii) a fixed number of values is corrupted for each input, and corruption is applied randomly [30]. All fault models involve multiple random bit flips. We then compute the Program Vulnerability Factor (PVF) [31], which represents the probability that a fault propagates through the model and manifests as Critical SDC.

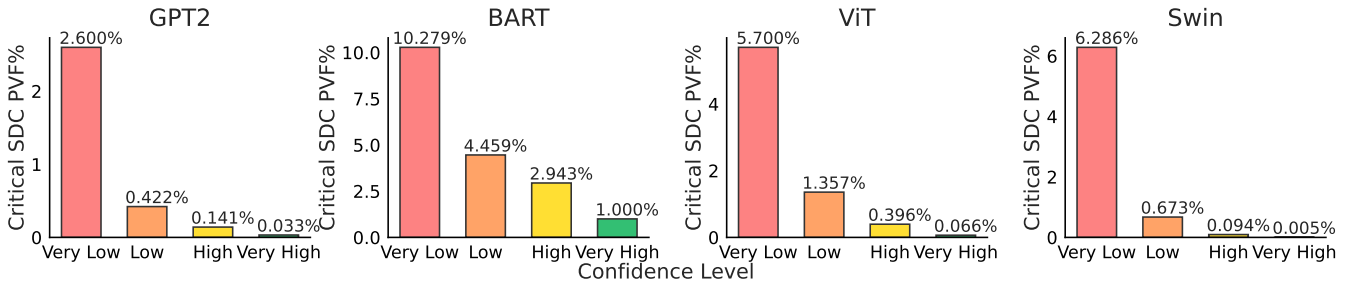

Fig. 4: Critical SDC PVF% for all assessed models, on different confidence intervals (Very Low, Low, High, and Very High).

To assess clipping effectiveness, we implement *per-layer activation clipping* for all models. The clipping range for each layer is computed using the profiling methodology of Section III. According to Burel *et al.* [6] and the activation distribution of Transformers models from Fig. 3, any activation value outside its profiled range is set to zero. Per-layer clipping is the most fine-grained configuration practical for Transformers. A more fine-grained alternative, such as per-activation clipping [9], incurs prohibitive overhead. A more coarse-grained approach (e.g., model-level range) necessarily increases the range, yielding weaker protection and more misclassifications.

B. Clipping effectiveness

Overall, despite applying per-layer activation clipping, it is insufficient to prevent Critical SDCs. For the entire validation dataset, the total Critical SDC PVF is 0.44% for GPT2, 1.31% for BART, 1.06% for ViT, and 0.41% for Swin. The most vulnerable operation differs across models. For GPT2, ViT, and Swin, the Block is the most affected, with 0.50–1.77% Critical SDCs. For BART, the MLP is the most impacted, reaching 4.45%. Critical SDCs correlate with the operation's respective ranges. For all models, the operation showing the highest Critical SDCs is also the one with the widest range, while operations with a narrower range consistently show lower Critical SDCs.

Even when all layers are protected by clipping, many corrupted activation values remain *within* the valid numerical bounds and continue propagating. Up to 1.31% of Critical SDCs originate from such inbound corruptions. This confirms that the intrinsic variability of Transformer activation ranges limits the effectiveness of fixed clipping ranges.

Insight 2: Corrupted activations within clipping ranges can still induce Critical SDCs

Per-layer activation clipping fails to filter all faulty activations because Transformer layers exhibit large and heterogeneous ranges. These inbound corruptions can induce up to 1.31% Critical SDCs. Critically, clipping becomes less effective as the range of activations widens.

Input Impact on Clipping Efficiency: The fault effects on DNN classification are heavily influenced by the model confidence [32]. We investigate how confidence affects the likelihood of a Critical SDC. For each input, we quantify confidence as the difference between the predicted class probability (`TOP1`) and the second-highest probability (`TOP2`). Inputs with confidence values close to zero indicate high uncertainty and are expected to be more sensitive to injected faults. To study this effect, we group the assessed inputs into four confidence intervals: Very Low (0–0.25), Low (0.25–0.5), High (0.5–0.75), and Very High (0.75–1).

Fig. 4 reports the Critical SDC PVF (in %) for each confidence interval for GPT2, BART, ViT, and Swin. Overall, all models exhibit the highest Critical SDC rates for Very Low confidence inputs.The Critical SDC PVF reaches 2.60% for GPT2, 10.28% for BART, 5.70% for ViT, and 6.29% for Swin. As confidence increases, the Critical SDC rate decreases. For Very High confidence inputs, the PVF of Critical SDC drops to 0.033% for GPT2, 1.000% for BART, 0.066% for ViT, and 0.005% for Swin. This trend occurs because low-confidence inputs require only small perturbations to change the `TOP1` prediction, while high-confidence inputs require much larger activation deviations to change the final class. This highlights a key limitation of activation clipping: *it is least effective precisely on the inputs where the model is most vulnerable*. A non-negligible portion of each dataset (Table I) falls into the Very Low and Low confidence categories.

Per-layer activation clipping can reduce extreme values, but cannot fully protect Transformers. Inputs with Very Low and Low confidence remain especially vulnerable, and operations with a wider activation range consistently show higher Critical SDC rates. As a result, many corrupted activations remain within the valid range and propagate through the model, revealing the inherent limitations of activation clipping across models and modalities.

Insight 3: Clipping is much less effective if the model is not confident

Low-confidence inputs are inherently more sensitive to faults [32]. Small perturbations within valid activation ranges can change the predicted class, leaving thousands of inputs vulnerable even when per-layer activation clipping is applied.

V. Final Remarks and Future Directions

In this work, we discussed the limitations of activation clipping for Transformers, as they exhibit significant variations in activation ranges and distributions across layers performing the same type of computation. Using large-scale SWFI, we

assessed the reliability of four representative Transformer models. Even with per-layer clipping, corrupted activation values can remain within valid numerical bounds. These inbound corruptions propagated through the network and induced misclassification, indicating that fixed numerical thresholds cannot reliably filter all faulty activations in models with wide and heterogeneous ranges. Our analysis further reveals that activation clipping fails to protect the most vulnerable inputs, those with low confidence, which represent a non-negligible fraction of the evaluated datasets.

Our findings indicate that activation clipping alone offers limited protection for Transformers. Future work will focus on improving activation clipping for Transformers by developing finer-grained mitigation strategies relying on both model architecture and input characteristics.

Acknowledgments and Artifacts

This work was partially supported by ANR FASY (ANR-21-CE25-0008-01), ANR RE-TRUSTING (ANR-21-CE24-0015-02), and the INRIA/UFRGS TCHE projects. To support reproducibility, all experiment code and data are available at https://github.com/lucasrqt/Transformers-FI.

References

[1] P. Rech, "Artificial neural networks for space and safety-critical applications: Reliability issues and potential solutions," *IEEE TNS*, 2024.

[2] M. H. Ahmadilivani *et al.*, "A systematic literature review on hardware reliability assessment methods for deep neural networks," *ACM Computing Surveys*, 2024.

[3] G. Li *et al.*, "Understanding error propagation in deep learning neural network (dnn) accelerators and applications," in *SC*, 2017.

[4] L.-H. Hoang *et al.*, "Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation," in *DATE*, 2020.

[5] Z. Chen *et al.*, "A low-cost fault corrector for deep neural networks through range restriction," in *IEEE/IFIP DSN*, 2021.

[6] S. Burel *et al.*, "Improving dnn fault tolerance in semantic segmentation applications," in *IEEE DFT*, 2022.

[7] L. Iurada *et al.*, "Transient fault tolerant semantic segmentation for autonomous driving," in *UNCV*, 2024.

[8] L. Roquet *et al.*, "Cross-layer reliability evaluation and efficient hardening of large vision transformers models," in *ACM/IEEE DAC*, 2024.

[9] B. Ghavami *et al.*, "Fitact: Error resilient deep neural networks via fine-grained post-trainable activation functions," in *DATE*, 2022.

[10] A. Radford *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, 2019.

[11] M. Lewis *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*, 2020.

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

[13] K. Ma *et al.*, "Error resilient transformers: A novel soft error vulnerability guided approach to error checking and suppression," in *IEEE ETS*, 2023.

[14] F. Libano *et al.*, "Selective hardening for neural networks in fpgas," *IEEE TNS*, 2019.

[15] X. Xue *et al.*, "Soft error reliability analysis of vision transformers," *IEEE VLSI*, 2023.

[16] U. Zahid *et al.*, "Fat: Training neural networks for reliable inference under hardware faults," in *IEEE ITC*, 2020.

[17] M. Abumandour *et al.*, "Weightsentry: Real-time bit-flip protection for deep neural networks on gpus," in *ACM HASP*, 2025.

[18] E. Ozen *et al.*, "Boosting bit-error resilience of dnn accelerators through median feature selection," *IEEE TCAD*, 2020.

[19] G. Esposito *et al.*, "Enhancing the reliability of split computing deep neural networks," in *IEEE IOLTS*, 2024.

[20] M. A. Hanif *et al.*, "Faq: Mitigating the impact of faults in the weight memory of dnn accelerators through fault-aware quantization," in *IJCNN*, 2023.

[21] Y. Jiang *et al.*, "Adfq-vit: Activation-distribution-friendly post-training quantization for vision transformers," *Neural Networks*, 2025.

[22] L. Chen *et al.*, "Q-dit: Accurate post-training quantization for diffusion transformers," in *IEEE/CVF CPVR*, Jun. 2025.

[23] A. Williams *et al.*, "A broad-coverage challenge corpus for sentence understanding through inference," in *NAACL*, 2018.

[24] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[25] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF ICCV*, 2021.

[26] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *IEEE CVPR*, 2009.

[27] G. Papadimitriou *et al.*, "Demystifying the system vulnerability stack: Transient fault effects across the layers," in *ISCA*, 2021.

[28] Y. Sun *et al.*, "Demystifying the resilience of large language model inference: An end-to-end perspective," in *SC*, 2025.

[29] C. Bolchini *et al.*, "Fast and accurate error simulation for cnns against soft errors," *IEEE TC*, 2022.

[30] F. F. d. Santos *et al.*, "Revealing gpus vulnerabilities by combining register-transfer and software-level fault injection," in *IEEE/IFIP DSN*, 2021.

[31] V. Sridharan *et al.*, "Eliminating microarchitectural dependency from architectural vulnerability," in *IEEE HCPA*, 2009.

[32] A. Mahmoud *et al.*, "Optimizing selective protection for cnn resilience," in *IEEE ISSRE*, 2021.