Global Clipper: Enhancing Safety and Reliability of Transformer-based Object Detection Models

Qutub Syed[1,3], Michael Paulitsch[1], Karthik Pattabiraman[2], Korbinian Hagn[1], Fabian Oboril[1], Cornelius Buerkle[1], Kay-Ulrich Scholl[1], Gereon Hinz[3] and Alois Knoll[3]

[1] *Intel Labs, Munich, Germany*
[2] *University of British Columbia, Vancouver, Canada*
[3] *Technical University of Munich, Munich, Germany*

Abstract

As transformer-based object detection models progress, their impact in critical sectors like autonomous vehicles and aviation is expected to grow. Soft errors causing bit flips during inference have significantly impacted DNN performance, altering predictions. Traditional range restriction solutions for CNNs fall short for transformers. This study introduces the Global Clipper and Global Hybrid Clipper, effective mitigation strategies specifically designed for transformer-based models. It significantly enhances their resilience to soft errors and reduces faulty inferences to 0%. We also detail extensive testing across over 64 scenarios involving two transformer models (DINO-DETR and Lite-DETR) and two CNN models (YOLOv3 and SSD) using three datasets, totalling approximately 3.3 million inferences, to assess model robustness comprehensively. Moreover, the paper explores unique aspects of attention blocks in transformers and their operational differences from CNNs.

1. Motivation

The adoption of Deep Neural Networks (DNNs) has significantly impacted various sectors, including autonomous vehicles [1], aviation, healthcare [2], and space exploration [3], where high safety and reliability are crucial. This has spurred the growth of computer vision research communities focused on safe AI, tackling areas such as out-of-distribution detection [4], adversarial robustness and model interoperability [5]. A DNN-based computer vision model processes images to classify objects and predict their bounding boxes.

Errors during inference can lead to faulty bounding boxes, significantly altering system behaviour and underscoring the critical need for safer hardware for model execution. DNN accelerators execute models at a high level by constructing a computational graph that uses General matrix-to-matrix multiplication (GEMM) [6] for extensive layer input and weight multiplications. Key components in this process are the Multiply-accumulate (MAC) units within the lower accelerator levels shown in fig. 1 [7]. MAC units in DNN accelerators lack ECC protection, making them particularly vulnerable to soft errors—a major reliability concern. Such errors, often caused by radiation, chip ageing, manufacturing variations, or thermal issues [8, 9, 10], can alter intermediary computational values, leading to incorrect inferences. Research shows that the soft error rate will increase with higher transistor density, reduced feature sizes, and more cores [11, 12, 13]. For example, a 100-core system of 16nm node may fail every 1.5 hours due to soft errors [11], significantly affecting predictions as these propagate through layers, as shown in fig. 1. Although soft errors do not cause permanent damage, they can result in substantial reliability degradation.

This paper proposes a technique for mitigating soft errors in object detection models at the application level. We simulated soft errors as bit flips using PytorchALFI [14], an open-source tool that integrates large-scale fault injection capabilities with PyTorch.

Range restriction solutions effectively mitigate soft errors in CNN-based DNN models by applying precalculated bounds at every activation layer, computed using 20% of validation images to determine the minimum and maximum restrictions [15]. However, current protective measures fall short against soft errors in transformer-based vision models due to the complexity of their architectures. Our analysis shows that existing solutions are inadequate, necessitating significant enhancements in error mitigation strategies for these advanced systems. Without such improvements, the robustness of transformer-based models is compromised, highlighting the urgent need for more sophisticated and tailored protection mechanisms. Transformer models [16, 17], characterized by their self-attention and large

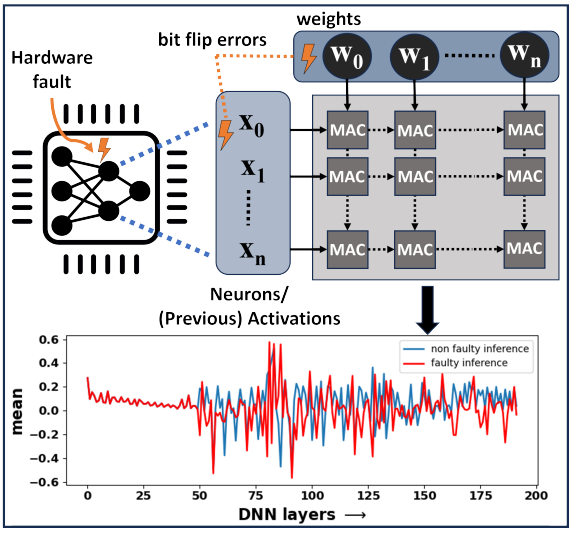


Figure 1: Abstract architecture of a DNN accelerator. The upper figure illustrates potential soft errors resulting in bit flips within neurons or weights at specific layers of the DNN model. The lower figure displays the mean values of layers in non-faulty inferences compared to faulty inference values when a bit-flip error is injected at the 50th layer of a transformer model DINO-DETR.

linear layers, are particularly vulnerable, as bit flip errors can cascade and significantly alter predictions.

For example, injecting a single-bit flip error into the 50th self-attention layer of the CoCo-trained DINO-DETR model, with 48M parameters [16], results in faulty inference such as ghost objects as shown in fig. 2b. These errors, which either create random high-confidence detections or erase them, can disrupt systems dependent on these models for tasks like object tracking, as shown in [18]. This underscores the significant impact of minor errors in complex networks. However, applying existing range restrictions cannot mitigate all ghost objects, as illustrated in fig. 2c.

We propose the Global Clipper and Global Hybrid Clipper range restriction layers as a straightforward yet vital enhancement to mitigate the impacts of soft errors in complex transformer-based models. These layers are implemented within the activation and linear layers of self-attention blocks, crucial points vulnerable to errors that can drastically affect network performance. This strategy involves a nuanced balance: preserving the network's ability to process diverse data inputs while ensuring robustness against errors that could lead to significant inaccuracies in outputs. By adding these range restriction layers, Global Clipper effectively safeguards the network's functionality without compromising its learning capabilities, ensuring sustained high performance even under challenging conditions. In the example, when Global Clipper is added, all the false ghost objects created by fault injection are removed as shown in fig. 2d.

In summary, the main contributions of this paper are as follows:

1. We introduce the Global Clipper and Global Hybrid Clipper fault mitigation techniques for transformer-based object detection models (section 4).
2. We present a comprehensive study with fault injection experiments across CNN and transformer models using three datasets, totalling 3.3 million inferences, to analyze vulnerabilities in these vision systems. We show that the proposed techniques are effective in reducing error rates from 6% to nearly 0% (section 5.3).
3. We explore the unique characteristics of attention blocks in transformers versus CNNs, providing insights into model vulnerabilities essential for enhancing safety throughout the life-cycle of deployed transformer models (section 5.4).

The following sections of the paper will first explore established methods for addressing soft errors in transformer-based DNN models. Then, the paper will detail the fault injection models under consideration (as discussed in above contributions - item 2). Next, our proposed solution, Global Clipper (item 1), will be introduced, emphasizing its effectiveness in mitigating soft errors. Subsequently, a thorough ablation study on implementing Global Clipper will be conducted(item 2). Finally, an examination and comparison of the vulnerability characteristics of Transformers and CNN across diverse datasets will be provided (item 3).

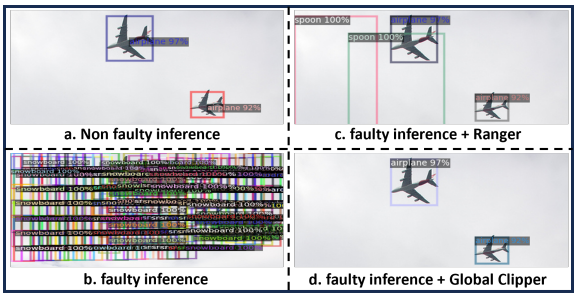


Figure 2: Visual example of faulty inferences on CoCo trained DINO-DETR model due to bit-flips caused by the soft errors.

2. Related Work

The reliability of safety-critical DNN models is assessed through various metrics at the application and hardware levels, enhancing their safety and reliability [19, 20, 15, 18]. Research has shown that DNNs are prone to soft errors, with instances of single-bit flips leading to faulty inferences [21, 22, 23, 18]. Traditionally, protection from soft errors in hardware has primarily involved error detection or correction codes (EDC or ECC) [24] for memory and using residuals for computing. These mechanisms are commonly implemented in high-end server-grade CPUs but less so in GPUs due to cost considerations or typical relaxed application requirements. Other techniques based on redundancy, like DMR (dual modular redundancy) and TMR (triple modular redundancy), are also used. Despite these measures, accelerators running DNNs may lack inherent protection. Furthermore, modular redundancy techniques can be implemented through ensembles, as discussed by [25], which may be used to detect and mitigate faults. However, these methods come with substantial computational overhead. To address this issue, budding ensemble solutions [26, 27] could be explored to reduce the computing overhead. Despite their potential, these solutions have not yet been demonstrated for mitigating or detecting soft errors.

Typical solutions for matrix operations in software, like algorithm-based fault tolerance (ABFT), have been adapted for DNNs but are limited by the overhead of checking large matrix multiplications typical in DNN applications [28, 29]. Researchers have also developed DNN-specific solutions at the application level based on range restriction solutions at the software level, particularly for CNN models, and explored using activation patterns to detect soft errors [30, 15, 29]. A small machine-learning module, reduced in dimension, analyses these patterns to identify and reject erroneous inferences [31, 32, 33, 34]. However, these methods face challenges related to scalability and complexity.

The vulnerability of DNNs to soft errors, including a significant number of CNN and few transformer-based models, is well-documented [10, 20, 18, 35, 36, 22]. However, previous studies have not extensively explored transformer models in object detection or conducted detailed, large-scale fault injection studies [37, 36]. This study aims to fill this gap by examining the resilience of transformer architectures and exploring effective mitigation strategies against soft

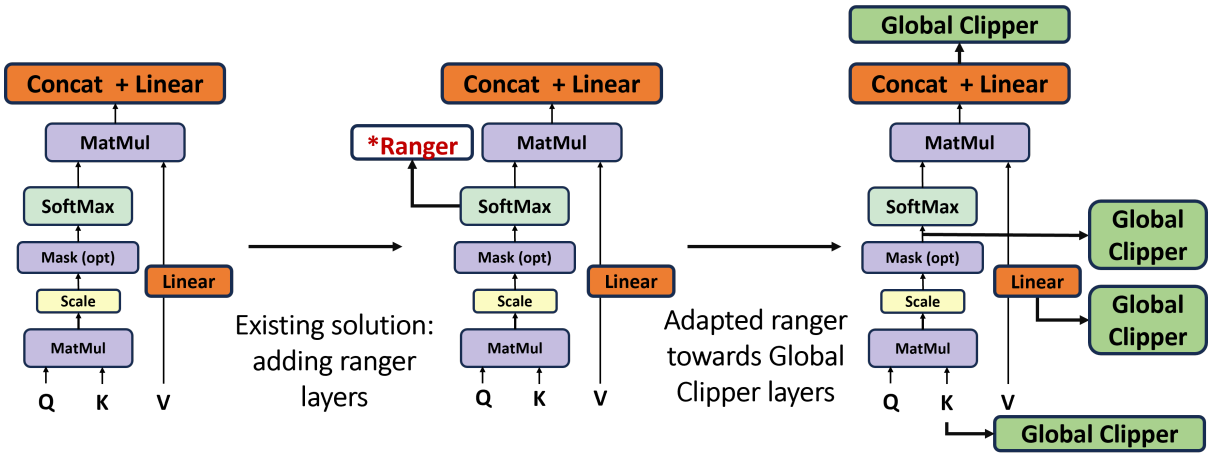


Figure 3: Integrating Global Clipper layers into transformer-based object detection models' self-attention blocks. *Ranger layers are recommended to be added to activation functions, usually at ReLU layers, not SoftMax.

errors in these architectures.

3. Fault Models

We consider soft errors in AI hardware accelerators, focusing on their impact on system reliability. These errors, typically manifesting as single or multiple-bit flips, can compromise data integrity by altering the model's weights and neurons, potentially skewing computations and decisions. Such disruptions in deep neural network operations are illustrated in fig. 1, highlighting the need for robustness strategies.

Parity or Error-Correcting Code (ECC) protects memory against soft errors, particularly crucial for essential memory blocks due to the significant overhead [38]. While ECC, especially SECDED (Single Error Correct, Double Error Detect) code, can detect and correct single-bit errors, it is limited to detecting two-bit errors without correction. This underscores the necessity for other techniques beyond ECC, where minimizing multi-bit errors is essential.

Our experimental setup injects faults as single or 10-bit flip errors during inference, with isolated injections in either the neurons or weights of the model, but not both simultaneously. This method ensures targeted and straightforward fault analysis, with each inference undergoing a single fault alteration. All models in our study employ 32-bit data types.

4. Global Clipper

Range restriction solutions [15, 30] effectively address bit flips caused by soft errors in CNN-based models by focusing on activation layers where convolutional layers attend to local image areas. This containment of deviations within localized feature map areas helps prevent extensive errors. However, these methods are less effective for transformer models, which employ global attention mechanisms across extensive linear layers [39]. In transformers, a bit flip can propagate errors throughout the multi-head attention layers, significantly altering vector representations and impacting predictions. This necessitates different mitigation strategies tailored to the global processing nature of transformers.

We introduce a crucial enhancement to existing range restriction layers, as illustrated in fig. 3, extending value monitoring and truncation from activation layers to linear layers within self-attention blocks. This strategy bolsters transformer architectures against soft-error-induced bit flips. The Global Clipper truncates out-of-range values to a predefined interval before deployment, operating at any activation or linear layer with bounds $B_{\text{lower}}, B_{\text{upper}}$, as detailed in eq. (1). Similarly, the Global Ranger restricts values within specified bounds, ensuring all layer outputs adhere to expected ranges.

These layers can be seamlessly fused at the application level, ensuring minimal overhead. Determining upper and lower bounds follows the approach outlined in previous range restriction solutions like Ranger [15]. Specifically, these bounds are computed using 20% of the training dataset, encompassing all activation and linear layers within transformer-based models as shown in the fig. 4.

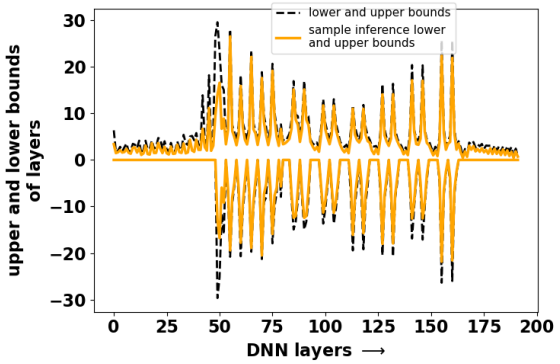
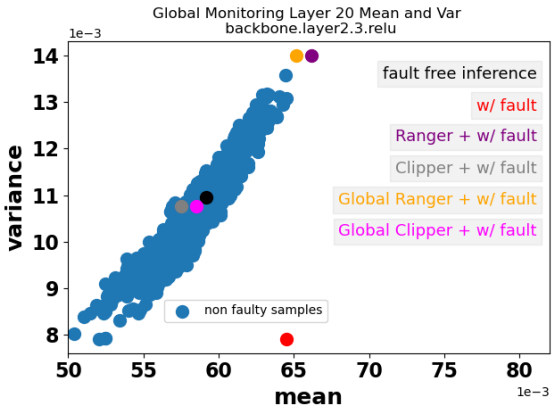
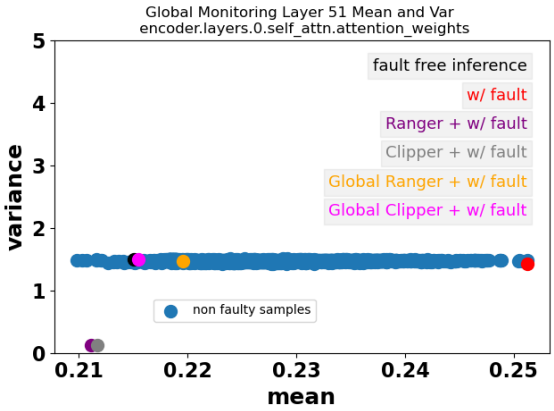


Figure 4: Lower and upper bounds for range restrictions, encompassing activation layers and linear layers within the self-attention blocks of the DINO-DETR model, are defined by the Global Clipper technique.



(a) Fault injected in ReLU layer



(b) Fault injected in linear layer

Figure 5: Tracking the mean and variance of layers within the DINO-DETR model, this illustration focuses on the ReLU activation layer and linear layer within the self-attention block.

$$L_{global\_clipper}.x/ = \begin{vmatrix} 0 & \text{if } x < B_{lower} \text{ or } x > B_{upper} \\ 0 & \text{if Inf â NaN} \\ x & otherwise \end{vmatrix} \quad (1)$$

We demonstrate the effectiveness of the Global Clipper solution using two experiments that involve injecting faults into the convolution and linear layers of existing models alongside our proposed method. Using the sample images from validation datasets, we introduce bit flip errors at a random MSB position in the second convolution layer of the ResNet50 backbone within the DINO-DETR model. We then monitor the mean and variance at the 20th ReLU activation layer and the 51st linear layer in the self-attention block immediately following the ResNet50 encoder. This process, illustrated in fig. 5, allows us to evaluate the performance of the Global Clipper and compare it with other mitigation strategies.

In the first experiment, we inject faults into the ReLU activation layer and apply various mitigation solutions, as shown in fig. 5(a). The Ranger method [15] confines data points within the extrapolated population cluster space. In contrast, the Clipper [30], along with the proposed Global Clipper, more tightly constrains the data to the region of non-faulty inferences. Although all techniques maintain acceptable tolerances, Ranger and Clipper tend to shift data points further from the original non-faulty positions at linear layers.

In our second experiment, we inject faults into the linear layer and activate various mitigation solutions. However, neither Ranger nor Clipper can confine the values to their original positions due to the sensitivity of self-attention layers to faults. Both Ranger and Clipper typically apply restrictions only at the activation layer; however, self-attention includes a SoftMax layer that cannot be similarly restricted without impairing the functionality and accuracy of the block. Hence, introducing the Global Clipper, shown in fig. 3, is essential for protecting attention blocks from faults. With faults introduced into the 51st linear layer, the Global Clipper successfully maintains data points closer to their original positions, as demonstrated in fig. 5(b). While the definition of Global Clipper resembles Ranger's, refining the bounds is crucial as it ensures that feature values remain within the sampled space in specific layers. In both experiments mentioned above, fault injection demonstrated using sample data doesn't necessarily result in faulty predictions. However, fault locations are sampled to visualize all data points within the area, providing a clear explanation of Global Clipper. Additionally, more faults affect the layer values, causing them to shift into log space compared to the fault-free visualizations. There are a few cases in which Global Clipper may not function out of the box on certain models, requiring slight modifications. These exceptions and adaptations are further elaborated upon in section 5.3.

5. Experiments

5.1. Experimental Setup

As introduced in section 1, soft errors, characterized by transient bit flip errors at the application level, impact individual inferences and last only until the next data fetch from memory. Bit flips at the sign and most significant bits of the mantissa minimally affect value ranges; however, flips at the sign and exponent bits of IEEE 754 floating-point arithmetic can alter predictions, which does not significantly change when testing other formats like BFloat16 as seen in [21, 18].

Our study assesses vulnerabilities in two transformer-based models, DINO-DETR [16] and Lite-DETR [17], and two CNN models, YOLOv3 [40] and SSD [41], across the CoCo [42], KITTI [43], and BDD100K datasets [44]. We conducted over 64 experiments, totalling about 3.3 million inferences. Each model undergoes experiments with random and targeted fault injections across all linear layers of self-attention blocks for transformers and convolution layers for CNN models.

Each data point extracted from these experiments includes 50,000 inferences from 1,000 image samples with random faults and 10,000 inferences per targeted fault experiment at each layer. Each experiment set repeats with baseline and proposed mitigation tech-

niques, like Clipper, Global Clipper or Global Hybrid Clipper, allowing a thorough analysis of model vulnerabilities and the effectiveness of mitigation strategies.

5.2. Evaluation Metrics

Accuracy metrics like AP50 or mAP [42] and their variants [45] are standard for evaluating fault injections in object detection models. Occasionally, these faults create ghost objects with lower confidence scores, not affecting the overall AP50 due to their exclusion in the PR curve area under curve (AUC) calculations [18]. To address this issue, the $IVMOD$ metric [18], which is insensitive to PR curve averaging, is employed. Faults that do not alter the model's outcome are considered benign. In contrast, significant faults are categorized into SDC (silent data corruption) and DUE (detectable and unrecoverable error) as recognized by the safety and reliability community [8].

By defining SDC and DUE [8] as critical faults, we enhance our ability to assess vulnerabilities effectively. This study introduces these conditions into the $IVMOD_{fd}$ metric for faulty detections (see eq. (2), eq. (3), and eq. (4)). Unlike the Global Clipper, the Global Ranger restricts values without truncating them. Vulnerability is evaluated by monitoring AP50 accuracy and $IVMOD_{fd}$. For example, if 30 out of 100 sampled images show detection discrepancies or encounter NaN or inf errors due to bit flips, the $IVMOD_{fd}$ would be 30%. Additionally, $IVMOD_{fd}$ can be used to estimate DNN accelerator vulnerability in terms of FIT rates [8] and other risk factors throughout the hardware's lifecycle [19, 20]. However, these aspects are beyond this paper's scope. Hereafter, $IVMOD_{fd}$ will be interchangeably called faulty detections. Moreover, in this study, $IVMOD_{fd}$ solely considers the 9 higher-order bits, including the sign and exponent bits, as described in section 3.

$$IVMOD_{SDC} = \frac{1}{N}\sum_{i=1}^{N}[.FP_{\text{orig}}/_i \text{ II } .FP_{\text{corr}}/_i \hat{a} \quad .FN_{\text{orig}}/_i \text{ II } .FN_{\text{corr}}/_i] \tag{2}$$

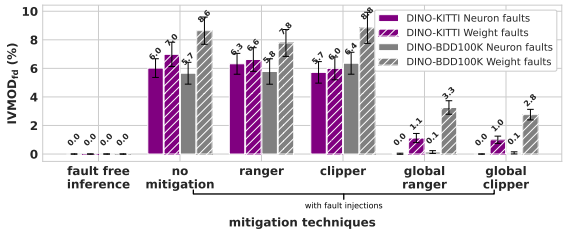
$$IVMOD_{DUE} = \frac{1}{N}\sum_{i=1}^{N}[\text{Inf } \hat{a} \text{ NaN}] \tag{3}$$

$$IVMOD_{fd} = IVMOD_{SDC}\, \hat{a}\ IVMOD_{DUE}, \tag{4}$$

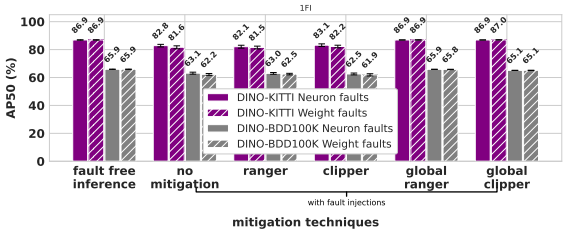
5.3. Results: Global Clipper on Transformer models

This section showcases the superior mitigation capability of Global Clipper and Global Hybrid Clipper over existing solutions like Ranger and Clipper. This is demonstrated on DINO-DETR and Lite-DETR models with various datasets injected with single bit-flip errors (see fig. 6 and fig. 7), we also investigate vulnerability across datasets, including layer-wise fault injection experiments.

The fig. 6 illustrates the model's vulnerability and evaluates the effectiveness of the various range restriction techniques in mitigating faults using $IVMOD_{fd}$



(a) Evaluating fault mitigation performance using $IVMOD_{fd}$ metric



(b) Evaluating fault mitigation performance using Average Precision metric

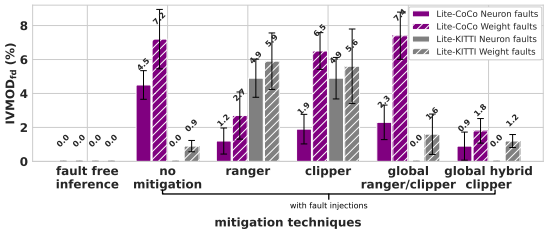
Figure 6: Evaluation of Global Clipper on DINO-DETR Transformer models.

and AP50 metrics. This includes results from experiments involving fault injections into weights and neurons. The Global Clipper significantly outperforms the existing solutions like Ranger and Clipper in mitigating the impact of the faults. The Global Clipper performs better in mitigating faults occurring in neurons, reducing faulty detections to nearly 0%, and for weight faults, the vulnerabilities are reduced to less than í 3% by outperforming other state-of-the-art algorithms like Ranger and Clipper. Additionally, as shown in fig. 6(b), a single bit flip in the inferences notably impacts the AP50 metric. For instance, the AP50 of DINO-DETR trained on KITTI decreases from 86.9 to 81.6 when injected with weight faults, and the Global Clipper effectively restores the AP50 to its original accuracy. In the context of DINO-DETR, Global Ranger and Global Clipper demonstrate similar mitigation performance. However, their performance is not as expected when applied to Lite-DETR due to the unique transformer architecture based on DINO-DETR. This presents an interesting scenario, leading to the introduction of the Global Hybrid Clipper. The Global Hybrid Clipper merges Global Clipper and Ranger layers. In this setup, Global Clipper is applied to Activation layers, while Global Ranger is used for the linear layers within self-attention blocks. In this scenario, the hybrid version of Global Clipper restores performance accuracy to baseline and reduces faulty detections significantly, as shown in fig. 7. For instance, it decreases from 4.5% to 0.5% in the case of neuron faults.

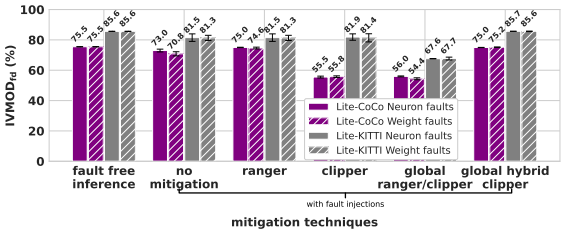
5.4. Ablation Study

5.4.1. Vulnerability and Resiliency Analysis of Attention and Convolution Layers

This section outlines the ablation study, including single-bit and multiple-bit (10-bit) flip experiments on



(a) Evaluating fault mitigation performance using $IVMOD_{fd}$ metric



(b) Evaluating fault mitigation performance using Average Precision metric

Figure 7: Evaluation of Global Clipper on Lite-DETR models.

CNN and transformer-based object detection models. Results are shown in fig. 9 and fig. 10. Global Clipper and Global Hybrid Clipper were employed for transformers, while Clipper was used for CNNs due to its superior performance over Ranger (see fig. 9(b) and fig. 10(b)).

Each plot represents 10,000 inferences, highlighting Global Clipper's superior handling of bit-flip errors. Comparing CNN and self-attention layers reveals that transformers generally exhibit greater fault-injection resilience than CNNs. For instance, DINO-DETR and Lite-DETR consist of six encoders and decoders, each with four linear layers (as depicted in fig. 8), resulting in 48 layers per model. These transformer layers exhibit distinct characteristics in response to fault injections compared to CNNs. These observations are consistent across both single-bit and multi-bit flip experiments (fig. 9 and fig. 10). Depending on the transformer model variant, encoders and decoders are interchangeable with self-attention blocks.

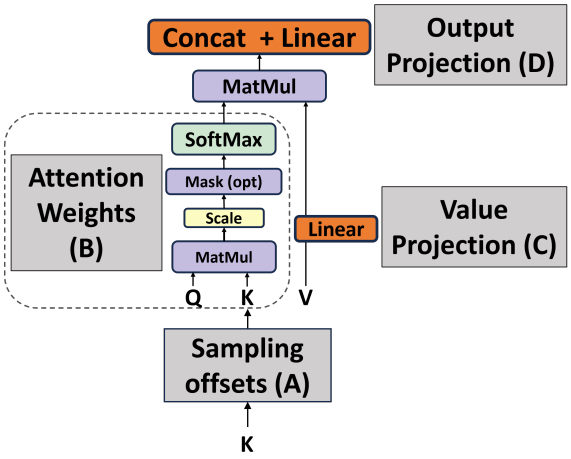


Figure 8: Individual components of Encoders and Decoders (Attention Blocks)

CNNs show no discernible pattern across layers, suggesting uniform susceptibility to generating faulty detections under bit-flip errors. In contrast, transformer models display greater inherent resilience. With suitable mitigation techniques, transformers offer enhanced safety against soft errors compared to CNN-based object detection models, making them preferable for applications demanding robustness against bit-flip errors.

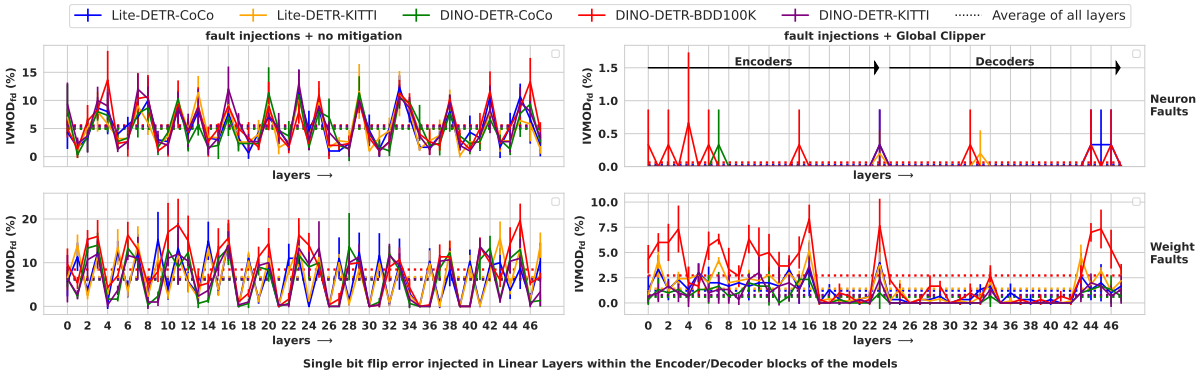
To better understand the transformer's vulnerability, we analyzed single-bit flip errors in attention block linear layers A, B, C, and D (see fig. 8). Data from fig. 9(a) and fig. 10(a) was segmented into four plots in fig. 11, illustrating each layer's vulnerability across encoders and decoders in 12 Attention Blocks. Layers such as Sampling Offset, Attention Weights, Value Projection, and Output Projection showed consistent vulnerability across CoCo, KITTI, BDD100K datasets, and DINO-DETR and Lite-DETR models for neuron faults. Variations in vulnerability due to weight faults depend on the attention block type, as shown in fig. 12. DINO-DETR uses Deformable Attention (DF) Layers, while Lite-DETR employs Key-Aware Deformable Attention (KDA) Layers, enhancing efficiency and attention mapping. The effects of weight faults on DF and KDA blocks vary, influencing encoders and decoders differently (fig. 12(b)), but neuron fault vulnerability remains stable across datasets and models (fig. 12(a)). These observations underscore two key transformer traits in vulnerability, distinct from CNNs:

- The vulnerability estimation of a transformer model's layers demonstrates consistent characteristics across different datasets during inference, unlike CNNs.
- The model's vulnerability estimation is influenced by the Self-Attention Block variant used in the model and remains consistent across different architectures and datasets during inference.

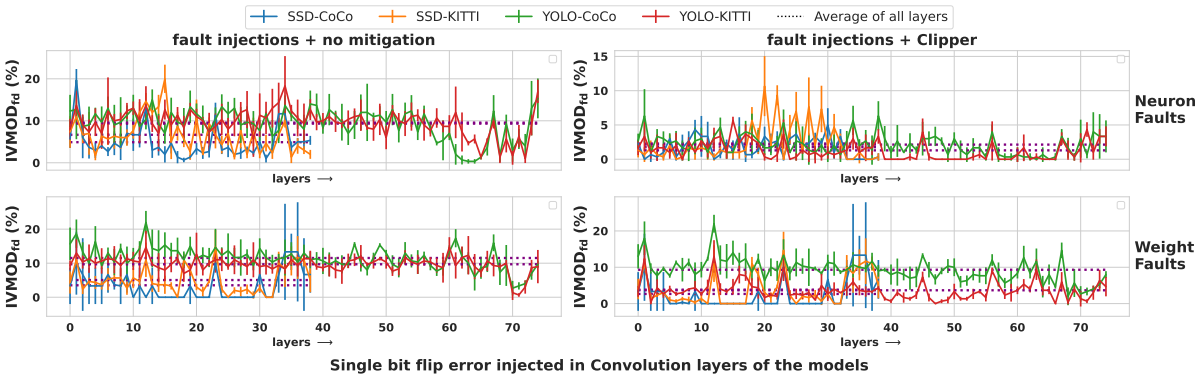
Thorough vulnerability analysis can greatly improve online safety and risk management, assisting in dynamic risk assessment and model monitoring across the lifecycle of deployed transformer-based models. Our findings demonstrate that despite continual learning and weight adjustments, these model's vulnerabilities remain stable, ensuring robust and reliable AI systems are maintained.

5.4.2. Integrating Global Clipper with minimal additional overhead

As depicted in fig. 3, the Global Clipper layers are integrated into four linear layers within the Attention Block (see fig. 8). The efficiency of adding these layers can be further evaluated by conducting a simple experiment that examines the impact of each Global Clipper Layer. This experiment entails the injection of a single-bit flip error at the initial stage of the Attention Block, particularly at the Sampling Offsets layer (stage A). Following this, the Global Clipper layer is selectively activated at different combinations of stages A, B, C, and D within the four linear layers of the

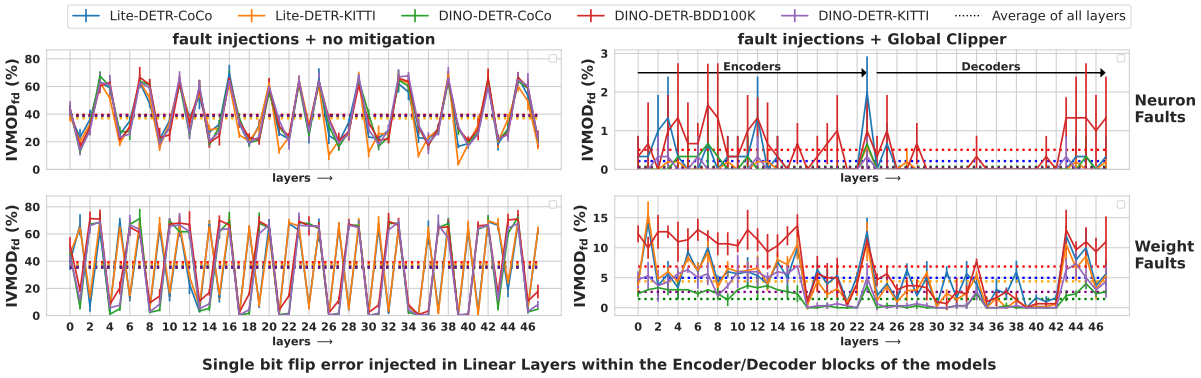


(a)

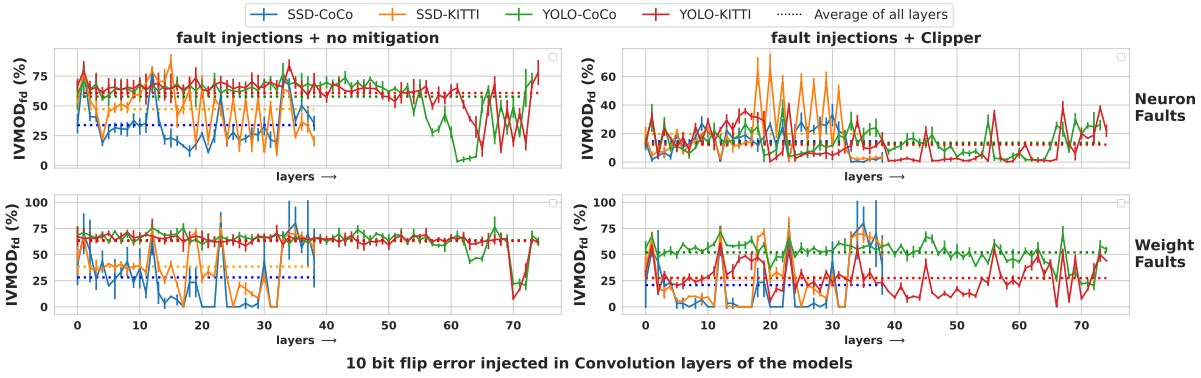


(b)

Figure 9: Layer-wise impact of Single-bit flip error on vulnerability metric based on faulty detections on transformers and CNN-based object detection models.

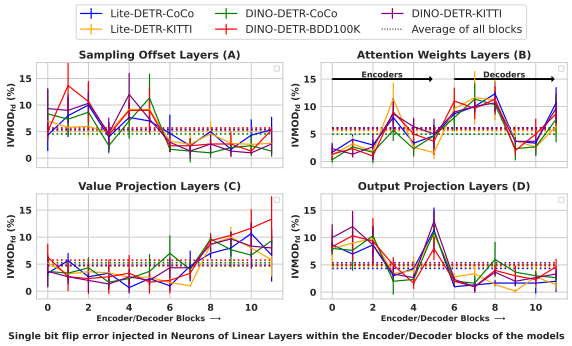


(a)

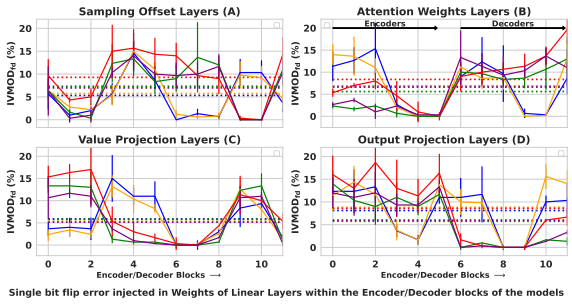


(b)

Figure 10: Layer-wise impact of 10-bit flip error on vulnerability metric based on faulty detections on transformers and CNN-based object detection models.



(a)



(b)

Figure 11: Impact of Single-bit flip error on individual linear layers of the Attention Blocks.

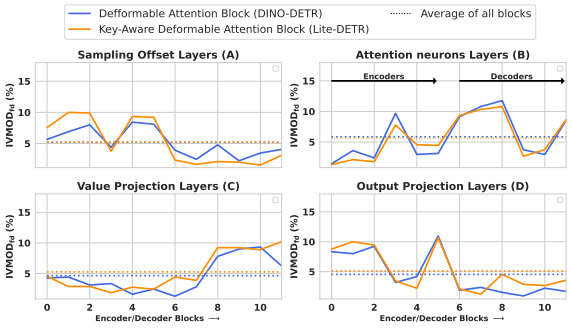
Attention Block, as detailed in fig. 8. Our observations indicate that integrating Global Clipper solely into the Output Projection and Value Projection layers (D and C in Figure 8), in addition to other activation layers, yields mitigation performance comparable to that achieved by incorporating it into all linear layers, as illustrated in Figure 13

6. Conclusion

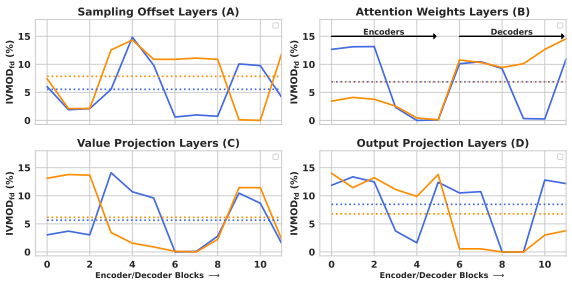
This study introduces Global Clipper and Global Hybrid Clipper to enhance the safety of transformer-based object detection models in critical settings, effectively minimizing faulty inferences to nearly 0%. We evaluated these solutions by conducting fault injection campaigns with transformer and CNN models across three datasets, totalling approximately 3.3 million inferences. Our extensive experiments and findings indicate that transformer models exhibit better inherent resilience to soft errors than CNN models. Evaluating these solutions provides insights into their effectiveness in real-world applications, contributing significantly to model robustness and computer vision safety. Future research should explore these solutions in transformer-based semantic segmentation and video tracking models to further enhance safety.

Acknowledgments

This work was partially funded by the Federal Ministry for Economic Affairs and Energy of Germany as part of the research project SafeWahr (Grant Number: 19A21026C).



(a) Neuron faults



(b) Weight faults

Figure 12: Comparison of vulnerability in Deformable Attention (DF) of DINO-DETR and Key-Aware Deformable (KDA) Attention in Lite-DETR under neuron and weight faults. The vulnerability metric ($IVMOD_{fd}$), shown in fig. 11, averaged across datasets, emphasizing differences between DF and KDA.

References

[1] J. Wang, L. Zhang, Y. Huang, J. Zhao, F. Bella, Safety of autonomous vehicles, Journal of advanced transportation 2020 (2020) 1–13.

[2] I. Habli, T. Lawton, Z. Porter, Artificial intelligence in health care: accountability and safety, Bulletin of the World Health Organization 98 (2020) 251.

[3] P. A. Oche, G. A. Ewa, N. Ibekwe, Applications and challenges of artificial intelligence in space missions, IEEE Access (2021).

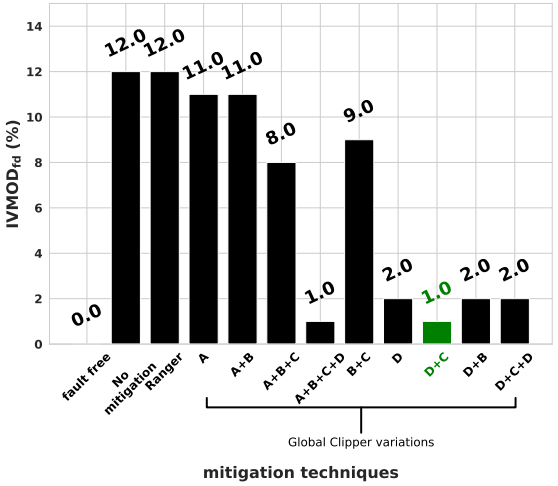


Figure 13: Integrating Global Clipper in Attention Block with minimal overhead.

[4] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al., A survey of uncertainty in deep neural networks, Artificial Intelligence Review 56 (2023) 1513–1589.

[5] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, Explainable ai: A brief survey on history, research areas, approaches and challenges, in: Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8, Springer, 2019, pp. 563–574.

[6] J. Gao, W. Ji, F. Chang, S. Han, B. Wei, Z. Liu, Y. Wang, A systematic survey of general sparse matrix-matrix multiplication, ACM Computing Surveys 55 (2023) 1–36.

[7] Y. Chen, Y. Xie, L. Song, F. Chen, T. Tang, A survey of accelerator architectures for deep neural networks, Engineering 6 (2020) 264–274.

[8] S. S. Mukherjee, J. Emer, S. K. Reinhardt, The soft error problem: An architectural perspective, in: 11th International Symposium on High-Performance Computer Architecture, IEEE, 2005, pp. 243–247.

[9] I. S. Haque, V. S. Pande, Hard data on soft errors: A large-scale assessment of real-world error rates in gpgpu, in: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010, pp. 691–696. doi:10.1109/CCGRID.2010.84.

[10] Y. Ibrahim, H. Wang, J. Liu, J. Wei, L. Chen, P. Rech, K. Adam, G. Guo, Soft errors in dnn accelerators: A comprehensive review, Microelectronics Reliability 115 (2020) 113969.

[11] G. R. Upasani, Soft error mitigation techniques for future chip multiprocessors (2016).

[12] S. Borkar, Designing reliable systems from unreliable components: the challenges of transistor variability and degradation, Ieee Micro 25 (2005) 10–16.

[13] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, L. Alvisi, Modeling the effect of technology trends on the soft error rate of combinational logic, in: Proceedings International Conference on Dependable Systems and Networks, IEEE, 2002, pp. 389–398.

[14] R. Gräfe, Q. S. Sha, F. Geissler, M. Paulitsch, Large-scale application of fault injection into pytorch models-an extension to pytorchfi for validation efficiency, in: 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), IEEE, 2023, pp. 56–62.

[15] Z. Chen, G. Li, K. Pattabiraman, A low-cost fault corrector for deep neural networks through range restriction, in: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE, 2021, pp. 1–13.

[16] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, H.-Y. Shum, Dino: Detr with improved denoising anchor boxes for end-to-end object detection, arXiv preprint arXiv:2203.03605 (2022).

[17] F. Li, A. Zeng, S. Liu, H. Zhang, H. Li, L. Zhang, L. M. Ni, Lite detr: An interleaved multi-scale encoder for efficient detr, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18558–18567.

[18] S. Qutub, F. Geissler, Y. Peng, R. Gräfe, M. Paulitsch, G. Hinz, A. Knoll, Hardware faults that matter: Understanding and estimating the safety impact of hardware faults on object detection dnns, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2022, pp. 298–318.

[19] A. Neale, M. Sachdev, Neutron radiation induced soft error rates for an adjacent-ecc protected sram in 28 nm cmos, IEEE Transactions on Nuclear Science 63 (2016) 1912–1917.

[20] G. Li, S. K. S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, S. W. Keckler, Understanding error propagation in deep learning neural network (dnn) accelerators and applications, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2017, pp. 1–12.

[21] F. Geissler, S. Qutub, S. Roychowdhury, A. Asgari, Y. Peng, A. Dhamasia, R. Graefe, K. Pattabiraman, M. Paulitsch, Towards a safety case for hardware fault tolerance in convolutional neural networks using activation range supervision, arXiv preprint arXiv:2108.07019 (2021).

[22] F. Geissler, S. Qutub, M. Paulitsch, K. Pattabiraman, A low-cost strategic monitoring approach for scalable and interpretable error detection in deep neural networks, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2023, pp. 75–88.

[23] A. Asgari Khoshouyeh, F. Geissler, S. Qutub, M. Paulitsch, P. Nair, K. Pattabiraman, Structural coding: A low-cost scheme to protect cnns from large-granularity memory faults, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2023, pp. 1–17.

[24] R. W. Hamming, Error detecting and error correcting codes, The Bell system technical journal 29 (1950) 147–160.

[25] N. Shlezinger, E. Farhan, H. Morgenstern, Y. C. Eldar, Collaborative inference via ensembles on the edge, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 8478–8482.

[26] S. S. Qutub, N. K. Cihangir, R. Rosales, M. Paulitsch, K. Hagn, F. Geissler, Y. Peng, G. Hinz, A. C. Knoll, Bea: Revisiting anchor-based object detection dnn using budding ensemble architecture., in: BMVC, 2023, pp. 792–797.

[27] S. S. Qutub, M. Paulitsch, K.-U. Scholl, N. K. Cihangir, K. Hagn, F. Oboril, G. Hinz, A. Knoll, Situation monitor: Diversity-driven zero-shot out-of-distribution detection using budding ensemble architecture for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2024, pp. 3502–3511.

[28] Z. Chen, Online-abft: An online algorithm based fault tolerance scheme for soft error detection in iterative methods, ACM SIGPLAN Notices 48

(2013) 167–176.

[29] K. Zhao, S. Di, S. Li, X. Liang, Y. Zhai, J. Chen, K. Ouyang, F. Cappello, Z. Chen, Ft-cnn: Algorithm-based fault tolerance for convolutional neural networks, IEEE Transactions on Parallel and Distributed Systems 32 (2020) 1677–1689.

[30] L.-H. Hoang, M. A. Hanif, M. Shafique, Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation, in: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2020, pp. 1241–1246.

[31] C. Schorn, L. Gauerhof, Facer: A universal framework for detecting anomalous operation of deep neural networks, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2020, pp. 1–6.

[32] C. Schorn, A. Guntoro, G. Ascheid, Efficient on-line error detection and mitigation for deep neural network accelerators, in: Computer Safety, Reliability, and Security: 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018, Proceedings 37, Springer, 2018, pp. 205–219.

[33] F. Zhao, C. Zhang, N. Dong, Z. You, Z. Wu, A uniform framework for anomaly detection in deep neural networks, Neural Processing Letters 54 (2022) 3467–3488.

[34] A. Mahmoud, S. K. S. Hari, C. W. Fletcher, S. V. Adve, C. Sakr, N. Shanbhag, P. Molchanov, M. B. Sullivan, T. Tsai, S. W. Keckler, Hardnn: Feature map vulnerability evaluation in cnns, arXiv preprint arXiv:2002.09786 (2020).

[35] X. Xue, C. Liu, Y. Wang, B. Yang, T. Luo, L. Zhang, H. Li, X. Li, Soft error reliability analysis of vision transformers, IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2023).

[36] L. Roquet, F. F. dos Santos, P. Rech, M. Traiola, O. Sentieys, A. Kritikakou, Cross-layer reliability evaluation and efficient hardening of large vision transformers models (2024).

[37] U. K. Agarwal, A. Chan, K. Pattabiraman, Resilience assessment of large language models under transient hardware faults, in: 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2023, pp. 659–670.

[38] A. Lotfi, S. Hukerikar, K. Balasubramanian, P. Racunas, N. Saxena, R. Bramley, Y. Huang, Resiliency of automotive object detection networks on GPU architectures, Proceedings - International Test Conference 2019-Novem (2019) 1–9. doi:10.1109/ITC44170.2019.9000150.

[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[40] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018).

[41] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, 2016, pp. 21–37.

[42] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[43] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 3354–3361.

[44] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, Bdd100k: A diverse driving dataset for heterogeneous multitask learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2636–2645.

[45] F. F. dos Santos, P. Navaux, L. Carro, P. Rech, Impact of reduced precision in the reliability of deep neural networks for object detection, in: 2019 IEEE European Test Symposium (ETS), IEEE, 2019, pp. 1–6.