Reliability of Deep Learning Models for Scanning Electron Microscopy Analysis

Chuen-Wun Pai, Hung-Wei Hsueh, Shu-han Hsu
Department of Computer Science
National Cheng Kung University
Tainan, Taiwan
{shhsu}@gs.ncku.edu.tw

Abstract

Scanning electron microscopy (SEM) provides high-resolution nanoscale imaging crucial for advanced materials characterization. Recent advancements in deep learning have significantly enhanced SEM analysis by automating the identification of nanoscale features. However, the reliability of these models remains insufficiently explored. We investigate a popular deep learning architecture, ResNet-50, by systematically injecting faults into their weight parameters for SEM characterization. Our analysis not only demonstrates how performance degrades under varying fault scenarios but also uncovers which layers and bit positions exhibit heightened vulnerability. These findings provide insights for developing robust fault-tolerant systems, including protective hardware measures and resilient model-training pipelines, thereby paving the way for more reliable deployment of SEM-based deep learning in industrial and research environments.

1 Introduction

Scanning electron microscopy (SEM) enables nanoscale visualization, indispensable in applications ranging from semiconductor inspection to microstructure identification. Deep learning has demonstrated potential in automating SEM analysis (Ede, 2021; Ge et al., 2020), enabling rapid identification of features for 0D nanoparticles (Kharin, 2020; Sun et al., 2022), 1D nanowires (Wong et al., 2021; Lin et al., 2022), 2D thin films (Modarres et al., 2017), and 3D patterns (Dey et al., 2022). However, despite the accuracy and efficiency of these deep learning systems, there is an important gap in understanding their reliability. In particular, faults or bit flips can compromise model outputs in ways that traditional training and evaluation pipelines do not address. This paper aims to bridge that gap by systematically investigating how single-event upsets affect deep learning models for SEM analysis, thus shedding light on new approaches for building robust, fault-tolerant systems.

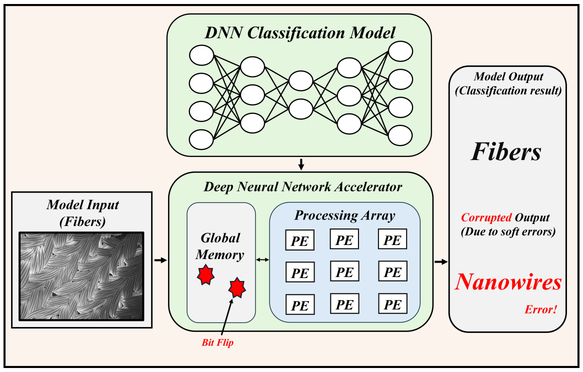


Figure 1: Example of a bit flip error leading to miscalculation and corrupted output.

Hard and soft errors are major sources of hardware unreliability, resulting in bit flips (Schroeder et al., 2009). Hard errors are permanent faults often from damage, manufacturing defects or wearout mechanisms (Li et al., 2008). Soft errors are often caused by radiation or transient disturbances (Slayman, 2011; Baumann, 2005). When bit flips occur during deep neural network (DNN) inference, they can lead to erroneous computations and ultimately degrade predictive accuracy, as shown in Fig. 1. In machine learning accelerators, two types of memory are typically used: (1) weight memory, storing trained network parameters, and (2) neuron (intermediate output) memory, which holds the outputs of each hidden layer (Neggaz et al., 2019). Flips in neuron memory affect only the current inference pass, whereas flips in weight memory persist until network redeployment, posing a critical risk over multiple inference rounds. We focus on weight memory errors, given their more significant impact.

In this work, we investigate the reliability of a popular deep learning model, ResNet-50, for SEM analysis through software-based fault injection (FI) methods. We adopt the single-precision IEEE 754 standard for weight representation, a prevalent convention in modern computing devices. This standard allocates 32 bits per floating-point value, with one sign bit (bit 31), eight exponent bits (bits 30–23), and 23 fraction bits (bits 22–0), thus providing a consistent foundation for evaluating error effects on network parameters.

Our contributions are as follows:

- **Reliability Assessment for SEM Analysis:** We utilize a fault-injection framework to assess deep learning analysis for SEM datasets, addressing a critical yet understudied domain in reliability research for materials characterization.
- **Layer- and Bit-Level Insights:** Through detailed fault injection experiments, we identify critical layers (e.g., initial convolutional layers) and bit positions (e.g., higher-order exponent bits) that exhibit high susceptibility to bit flips.
- **Practical Accuracy Impact:** We measure accuracy drops to gauge the impact of critical faults (bit flips causing output errors). Even a single fault can yield large inaccuracies, posing significant financial and scientific risks, particularly in industrial or research contexts.

By analyzing model performance under single-event upsets, we identify key vulnerabilities and provide practical insights for protecting these architectures in industrial and research contexts.

2 Related Works

Existing studies on fault injection (FI) in machine learning have largely focused on image classification (Ruospo et al., 2023; Arechiga & Michaels, 2018) and object detection (Qutub et al., 2022; Lotfi et al., 2019) tasks, using standard color-image datasets such as CIFAR-10, ImageNet, and Kitti. In image classification, the error rate can rise from 10% to 40% under hardware-induced faults, (Rahman et al., 2024), and permanent faults can lead to a 56% loss in accuracy (Siddique & Hoque, 2023). For object detection, permanent stuck-at faults may generate false positives covering as much as 83% of an image's area and miss up to 63% of true positives (Qutub et al., 2022).

Despite these insights, the reliability of machine learning models processing SEM analysis remains an open question. Scientific SEM images differ from real-world images in both format and content: they are high-resolution, grayscale representations of microstructures, surface morphology, and material composition, whereas real-world images highlight contextual features, colors, and semantic relationships. Given the precision requirements in industrial applications (e.g., semiconductor manufacturing) or scientific research, the cost of failing to detect or correctly identify features can be high. Thus, our work evaluates the resilience of deep learning models under fault injection for SEM analysis for robust deployment in nanoscale workflows.

3 Reliability analysis methodologies

3.1 Dataset and Model

To evaluate model reliability, we utilize the SEM dataset introduced by Aversa et al. (2018), consisting of an annotated collection of scanning electron microscopy images spanning 10 categories. For our experiments, we adopt a ResNet-50 architecture pretrained on the ImageNet dataset and apply transfer learning on the SEM dataset. Following the approach of Modarres et al. (2017), we retrain only the final fully connected layer while freezing all preceding layers. We split the dataset into 80% for training, 15% for validation, and 5% for testing, and train for 100 epochs using the Adam optimizer with a cross-entropy loss function. The test accuracy is 91%, with a precision of 91%, and a recall of 87% on the test set. This setup ensures a strong baseline for assessing the resilience of the deep learning model under single-event upsets in weight memory.

3.2 Fault injection methods

We focus on single-bit flips in model weights, treating any flip that causes misclassification on originally correctly classified SEM images as a critical bit flip error. To evaluate the impact of such faults, we adopt a fault injection (FI) workflow: (1) we train ResNet-50 on the SEM dataset, (2) we run inference on the test set to identify correctly classified images, (3) for each experiment, we inject a single bit flip into the model weights, and (4) we document whether this perturbation causes misclassification. We use PyTorchFI (Mahmoud et al., 2020) to simulate soft errors at the software level.

Although exhaustive FI offers the most thorough assessment by injecting faults in all bits for every parameter, the size of modern DNNs makes this approach infeasible (e.g., ResNet-50 would require over 9,000 days on a single NVIDIA RTX A4000 GPU). To overcome this limitation, we employ the statistical FI approach (Ruospo et al., 2023), which strategically samples a smaller subset of potential faults. This dramatically reduces computational demands, requiring around 30 days on the same hardware, yet still provides a robust approximation of fault behavior under practical assumptions.

The probability of inducing a critical failure is assumed to be constant for bit flips at the same bit position in a given layer's parameters, allowing each fault injection to be treated as a Bernoulli trial with fixed success probability. The parameter space N is divided into subpopulations $N(i, l)$, where i denotes the bit position and l the layer. Each subpopulation's critical failures follow a Binomial distribution, approximated by a normal distribution (per the Central Limit Theorem) for sufficiently large samples. Instead of testing all elements in N (exhaustive fault injection), we sample a smaller subset $n(i, l)$ from each subpopulation to estimate the critical failure rate. This approach substantially reduces computational overhead while maintaining statistically valid reliability estimates.

The statistical FI sample size is obtained by the following three formulas:

$$n(i,l) = \frac{N(i,l)}{1 + e^2 \cdot \frac{N(i,l)-1}{t^2 \cdot p(i,l) \cdot (1-p(i,l))}} \tag{1}$$

where $n(i, l)$ is the sample size of each subpopulation. The error margin, e, is set at 1%. $N(i, l)$ is the actual subpopulation size, and $p(i, l)$ is the probability that a bit flip induces a critical failure.

$$\forall i \in I, l \in L \qquad D_{avg}(i,l) = D_{0\to1}(i,l) \cdot f_0(i,l) + D_{1\to0}(i,l) \cdot f_1(i,l) \tag{2}$$

where $D_{0-1}(i, l)$ is the average distance between all the original and the faulty weights produced by a bit-flip from 0 to 1 on the i^{th} bit. $f_0(i, l)$ represents the number of times the i^{th} bit in layer l equals 0 in the weight distribution, and $f_1(i, l)$ represents the number of times the i^{th} bit equals 1.

$$\forall i \in I, l \in L \quad \to \quad p(i,l) = a + \frac{(D_{avg}(i,l) - D_{avg-min}(l)) \cdot (b-a)}{D_{avg-max}(l) - D_{avg-min}(l)} \tag{3}$$

$p(i, l)$ is calculated by performing a min-max normalization of D_{avg} between a = 0.01 and b = 0.99. Because the bit-flip distance for the 30th bit is significantly larger than the rest, we exclude it from normalization and set its probability to 0.99.

Unlike prior work (Ruospo et al., 2023), which studies bit-sticking faults, our studies also include soft errors, which may be induced by environmental factors such as radiation or electrical noise. Our fault injections target the weights of convolutional (Conv) and fully connected (FC) layers, which are the critical backbone for feature extraction and classification, thus ensuring a focused assessment of reliability critical components.

4 Experimental results

We perform statistical FI on our trained ResNet-50 model, analyzing results from three perspectives: layer-level, bit-level, and accuracy drop analysis. Note that the critical failure rate is defined as the proportion of injected faults leading to classification errors in images originally classified correctly.

4.1 Layer-level analysis

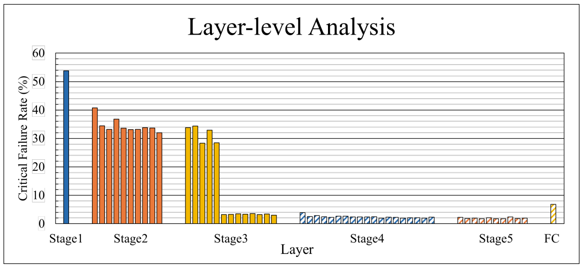


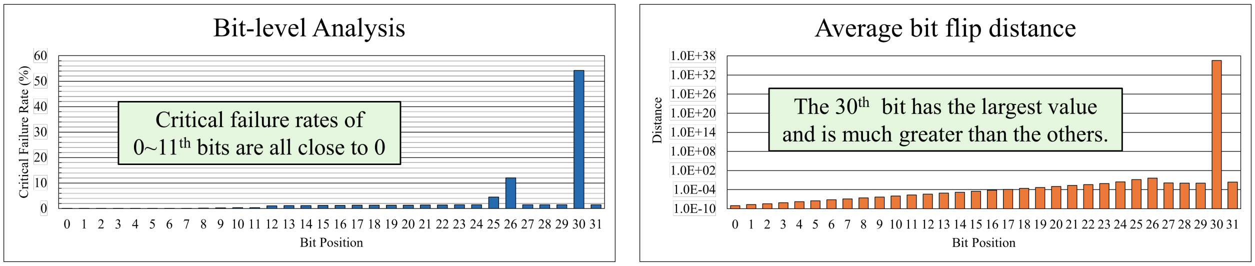
Figure 2: Critical failure rate vs. layer. Early-stage layers show higher failure rates than later layers.

ResNet-50 can be broadly divided into an initial convolution-plus-pooling stage, four sequential residual stages (each composed of multiple bottleneck blocks), and a final classification head (global average pooling followed by a fully connected (FC) layer). Because the network contains 53 convolutional layers (including the convolutional layer in downsample shortcuts), we group them for clarity as: the first stage (stage 1) and the four main stages (stage 2 to 5).

We find that stage 1 exhibits the highest vulnerability, with a critical failure rate exceeding 50%, as shown in Fig. 2. Stage 2 and the early layers of stage 3 also demonstrate elevated failure rates, frequently surpassing 30%. In the remaining sections of the network, fault susceptibility stabilizes at a comparatively lower level; however, a slight uptick is seen in the final fully connected layer.

Early-stage low-level features are progressively extracted and reassembled in subsequent convolutional layers to form high-level features. Therefore, if bit flips occur in the initial layers, these errors may propagate and accumulate through multiple convolutional layers, affecting the representation of high-level features and ultimately increasing the likelihood of an incorrect output.

4.2 Bit-level analysis



(a)

(b)

Figure 3: (a) Critical failure rate vs. bit position. (b) Flip distance vs. bit position.

To assess the robustness of bit positions, we measure their critical failure rates and correlate these with the average bit flip distance, shown in Fig. 3. The average bit flip distance is the numerical difference induced by flipping a given bit. As shown in Fig. 4, bits 0–11 (the lower half of the mantissa for the IEEE 754 single-precision format) exhibit near-zero failure rates, while bits 25, 26, and 30 display markedly higher rates (4%, 12%, and >50%, respectively). Notably, bit 30 imposes the largest flip distance by a substantial margin, leading to the highest observed failure rate; bits 25 and 26 also exhibit above-average flip distances, aligning with their elevated fault susceptibilities. This pattern indicates that larger flip distances are generally associated with higher corruption rates, highlighting the need for bit-specific fault tolerance mechanisms.

4.3 Accuracy drop analysis

To quantify the impact of critical failure inducing bit flips, we measure accuracy drop as the proportion of the test set misclassified when a single bit flip causes a critical fault. For instance, if a bit flip leads to 169 misclassifications out of 845 test images, the accuracy drop is 20%. In many industrial scenarios, a model is loaded once and used repeatedly;

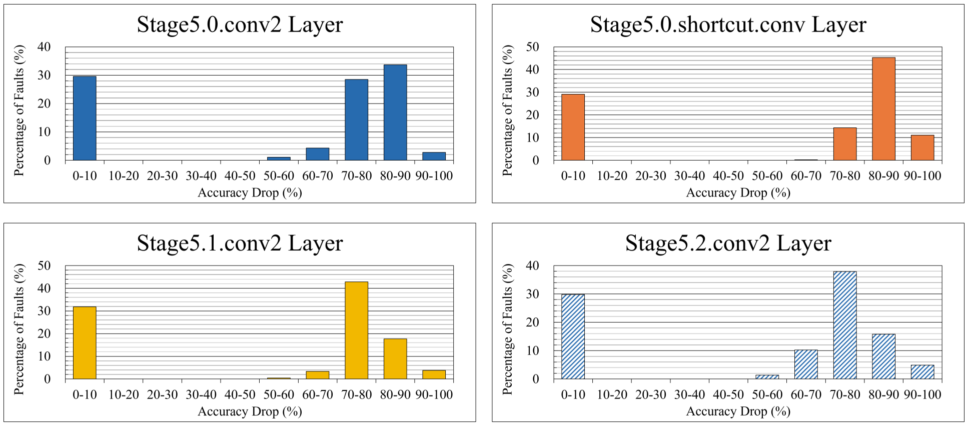


Figure 4: Layers causing significant accuracy drops.

a single bit flip causing a 50% accuracy drop would therefore affect 50% of products until the model is reloaded or corrected, posing substantial financial and operational risks.

Our analysis, shown in Fig. 4, reveals that certain layers within Stage 5 of ResNet-50 experience large accuracy drops, with over 70% of critical failures in these layers leading to a final accuracy drop above 50%. Note that for example, "Stage5.0.conv2" indicates the second convolution layer within the first (index 0) bottleneck block of the final (fifth) stage. These layers exhibit high-impact failures due to two main factors: (1) Stage 5 is tasked with extracting the highest-level features, which are pivotal for SEM classification, and (2) the conv2 (3×3) and shortcut conv layers in bottleneck blocks play critical roles in spatial feature extraction and direct feature propagation.

While earlier stages (e.g., Stage 1–2) handle lower-level features where critical bit flips may be more frequent, the disruptions generally do not directly alter key high-level features, and thus the overall accuracy drop is less severe. By contrast, errors in stage 5, though less frequent, can directly corrupt the core features needed for final classification, causing large drops in accuracy. Furthermore, in stage 5, the shortcut connections pass original features directly to the classifier, making any fault here immediately influential on the final output. Meanwhile, the conv2 layer is the primary 3×3 convolution, which handles the majority of spatial feature extraction in stage 5. Taken together, these findings highlight the importance of selective protection, such as error-correcting codes or hardware redundancy, in safeguarding the most vulnerable layers of ResNet-50 for SEM classification.

5 Conclusion

In this work, we conducted a systematic study on the resilience of deep learning models for a widely used architecture, ResNet-50, when subjected to single-bit flips in weight memory for scanning electron microscopy analysis. By leveraging a statistical fault injection framework, we identified which layers (e.g., initial convolution layers) and which bit positions (e.g., the 25th, 26th, and 30th) are most vulnerable. Our results reveal that faults in early-stage layers cause notably higher failure rates, and bit flips associated with larger flip distances yield disproportionately greater performance degradation. Furthermore, four layers (Stage5.0.conv2, Stage5.0.shortcut.conv, Stage5.1.conv2, Stage5.2.conv2) experience large accuracy drops, where over 70% of their failure-inducing bit flips result in a final accuracy drop exceeding 50%. These findings highlight the necessity for tailored reliability mechanisms, such as error-correcting codes or selectively protecting critical layers and bit positions, to ensure robust deployments of deep learning–driven SEM analysis systems in industrial and research environments.

Acknowledgments

Thank you to the reviewers for their detailed feedback allowing us to improve and clarify our work. We are also grateful for the support from the National Science and Technology Council (Taiwan, ROC) through Project NSTC 112-2221-E-006 -151 -MY3 and Google Research Scholar Award.

References

Austin P Arechiga and Alan J Michaels. The robustness of modern deep learning architectures against single event upset errors. In *2018 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–6. IEEE, 2018.

Rossella Aversa, Mohammad Hadi Modarres, Stefano Cozzini, Regina Ciancio, and Alberto Chiusole. The first annotated set of scanning electron microscopy images for nanoscience. *Scientific Data*, 5(1):1–10, 2018.

Robert C Baumann. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Transactions on Device and Materials Reliability*, 5(3):305–316, 2005.

Bappaditya Dey, Dipam Goswami, Sandip Halder, Kasem Khalil, Philippe Leray, and Magdy A Bayoumi. Deep learning-based defect classification and detection in sem images. In *Metrology, Inspection, and Process Control XXXVI*, pp. PC120530Y. SPIE, 2022.

Jeffrey M Ede. Deep learning in electron microscopy. *Machine Learning: Science and Technology*, 2(1):011004, March 2021.

M. Ge, F. Su, Z. Zhao, and D. Su. Deep learning analysis on microscopic imaging in materials science. *Materials Today Nano*, 11:100087, 2020.

Jacob Gildenblat and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

A Yu Kharin. Deep learning for scanning electron microscopy: Synthetic data for the nanoparticles detection. *Ultramicroscopy*, 219:113125, 2020.

Man-Lap Li, Pradeep Ramachandran, Swarup K. Sahoo, Sarita V. Adve, Vikram S. Adve, and Yuanyuan Zhou. Understanding the propagation of hard errors to software and implications for resilient system design. *ACM SIGPLAN Notices*, 43(3):265–276, 2008.

Binbin Lin, Nima Emami, David A Santos, Yuting Luo, Sarbajit Banerjee, and Bai-Xiang Xu. A deep learned nanowire segmentation model using synthetic data augmentation. *npj Computational Materials*, 8(1):88, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Atieh Lotfi, Saurabh Hukerikar, Keshav Balasubramanian, Paul Racunas, Nirmal Saxena, Richard Bramley, and Yanxiang Huang. Resiliency of automotive object detection networks on gpu architectures. In *2019 IEEE International Test Conference (ITC)*, pp. 1–9. IEEE, 2019.

Abdulrahman Mahmoud, Neeraj Aggarwal, Alex Nobbe, Jose Rodrigo Sanchez Vicarte, Sarita V Adve, Christopher W Fletcher, Iuri Frosio, and Siva Kumar Sastry Hari. Pytorchfi: A runtime perturbation tool for dnns. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 25–31. IEEE, 2020.

Mohammad Hadi Modarres, Rossella Aversa, Stefano Cozzini, Regina Ciancio, Angelo Leto, and Giuseppe Piero Brandino. Neural network for nanoscience scanning electron microscope image recognition. *Scientific Reports*, 7 (1):13282, 2017.

Mohamed A Neggaz, Ihsen Alouani, Smail Niar, and Fadi Kurdahi. Are cnns reliable enough for critical applications? an exploratory study. *IEEE Design & Test*, 37(2):76–83, 2019.

Syed Qutub, Florian Geissler, Yang Peng, Ralf Gräfe, Michael Paulitsch, Gereon Hinz, and Alois 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*, pp. 298–318. Springer, 2022.

Md Hasanur Rahman, Sabuj Laskar, and Guanpeng Li. Investigating the impact of transient hardware faults on deep learning neural network inference. *Software Testing, Verification and Reliability*, pp. e1873, 2024.

Annachiara Ruospo, Gabriele Gavarini, Corrado De Sio, J Guerrero, Luca Sterpone, M Sonza Reorda, Ernesto Sanchez, Riccardo Mariani, Joseph Aribido, and Jyotika Athavale. Assessing convolutional neural networks reliability through statistical fault injections. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6. IEEE, 2023.

Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber. Dram errors in the wild: A large-scale field study. *SIGMETRICS Performance Evaluation Review*, 37(1):193–204, 2009.

Ayesha Siddique and Khaza Anuarul Hoque. Exposing reliability degradation and mitigation in approximate dnns under permanent faults. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 31(4):555–566, 2023.

Charles Slayman. Soft error trends and mitigation techniques in memory devices. In *2011 Proceedings-Annual Reliability and Maintainability Symposium*, pp. 1–5. IEEE, 2011.

Zhijian Sun, Jia Shi, Jian Wang, Mingqi Jiang, Zhuo Wang, Xiaoping Bai, and Xiaoxiong Wang. A deep learning-based framework for automatic analysis of the nanoparticle morphology in sem/tem images. *Nanoscale*, 14(30): 10761–10772, 2022.

CH Wong, SM Ng, CW Leung, and AF Zatsepin. The effectiveness of data augmentation in porous substrate, nanowire, fiber and tip images at the level of deep learning intelligence. *arXiv preprint arXiv:2103.12526*, 2021.

A Appendix

A.1 Ballpark Estimate of Bit Flips in SEM Analysis

Hard and soft errors can both contribute to bit flips. For soft errors, electromagnetic interference and transient power fluctuations are typically the most likely significant disturbances affecting SEM. Standard SEMs generally run in controlled lab or industrial environments with shielding and vibration isolation, so ambient cosmic radiation is less likely to cause direct hardware failures. However, in cases where maintenance is lacking or shielding integrity has degraded, there exists a non-negligible possibility of radiation leakage from the SEM's own electron beam. In high-radiation environments (e.g., examining radioactive samples), SEM analysis electronics could face increased risk, but this scenario is not the norm for most SEM analysis deployments.

Although SEUs may seem less critical than issues such as dataset quality, model generalization, or hyperparameter tuning, their impact becomes increasingly relevant as datasets scale and models are deployed across a wider range of hardware environments. Our aim in highlighting SEUs is to draw attention to an often-overlooked dimension of reliability in deep learning systems. SEUs are particularly important to consider in radiation-prone environments (e.g., high-altitude systems), long-running inference pipelines where models remain in memory for extended periods, and deployment settings that lack frequent weight reloading or hardware-level error detection mechanisms.

Due to intellectual property restrictions, explicit hard and soft error rate data for GPU VRAM is not publicly available. However, VRAM is based on DRAM technology. According to a large-scale study (Schroeder et al., 2009), DRAM error rates (encompassing both soft and hard errors) range from approximately 25,000 to 70,000 failures-in-time (FIT) per megabit, equivalent to 25,000–70,000 errors per billion device hours per megabit. Although the study does not separately quantify soft and hard errors, it notes that approximately 70% of observed errors recur at the same memory address, suggesting they are likely hard errors. Soft errors, by contrast, tend to be non-deterministic and spatially random. From this, one can infer that soft errors constitute roughly 30% of the total error rate. Based on these estimates, a 1 GB memory module could experience approximately 1.47 to 4.13 soft errors per day, and 3.44 to 9.63 hard errors per day.

In typical SEM analysis configurations, total memory capacity may range from 12 GB to 32 GB, with newer systems often supporting even larger capacities. Thus, the number of both soft and hard errors increases proportionally with memory size and continuous uptime. Table 1 estimates the hard and soft errors rates for a typical SEM system.

Even if the overall flip probability is low, large model sizes, extended operating periods (e.g., semiconductor manufacturing that demand round-the-clock throughput), or infrequent weight reloading can make a single disruptive bit flip more likely over time. In high-value industrial or research contexts, a rare but high-impact failure can pose serious risks and be costly. Hence, our work seeks to highlight these potential vulnerabilities for SEM deep learning analysis.

Practitioners can utilize our resilience analysis methodology to make informed decisions balancing reliability, performance, and cost based on the specific needs of their SEM system. In cases where frequent model reloading is not feasible, such as during continuous operation for semiconductor manufacturing inspections or long experimental sessions, alternatives like in-memory verification or targeted redundancy in critical layers may offer more effective and economical solutions.

Table 1: Estimated Hourly Error Rates by Memory Size and Error Type in SEM

Error Type	Memory Size (GB)	SEU Rate Range (FIT/Mb)	Errors per Hour (Best Case)	Errors per Hour (Worst Case)
Soft	12	7500–21000 FIT/Mb	0.74	2.06
	16		0.98	2.75
	32		1.97	5.51
	64		3.91	11.01
Hard	12	17500–49000 FIT/Mb	1.72	4.82
	16		2.29	6.42
	32		4.59	12.85
	64		9.18	25.69

A.2 Reliability Analysis of Swin Transformer for SEM Classification

To further explore the reliability of models in material analysis, we adopt a Swin Transformer (tiny architecture) (Liu et al., 2021) pretrained on ImageNet and apply transfer learning to the SEM dataset. The Swin Transformer leverages a hierarchical Transformer framework with shifted window attention and has demonstrated strong performance in computer vision tasks. Using the same training hyperparameters as ResNet-50, we achieve a test accuracy of 93%, with 91% precision and 90% recall.

Swin Transformer can be broadly divided into an initial patch partition module, four main stages (the first stage contains a linear embedding module and Swin Transformer blocks, while the next three stages each contain a patch merging module followed by additional Swin Transformer blocks), and a final classification head consisting of Layer Norm followed by a fully connected layer.

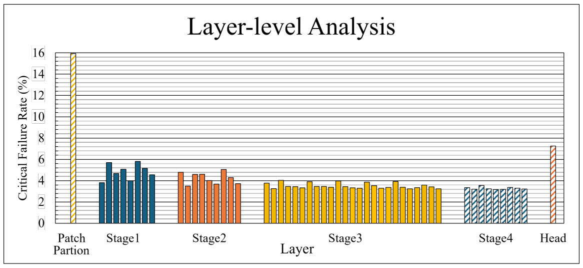
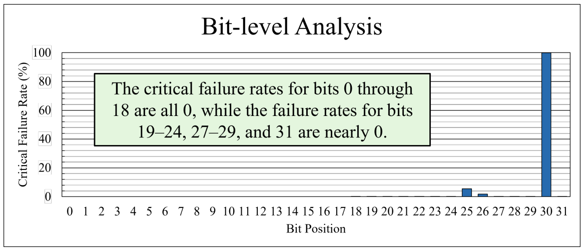


Figure 5: Critical failure rates across Swin Transformer layers. The patch partition module and the classification head exhibit the highest vulnerability, while the main body (stages 1–4) remains comparatively stable.

In our bit-flip experiments, we target the parameters of convolutional and fully connected layers. The layer-level robustness analysis (Fig. 5) shows that the patch partition module and the classification head exhibit the highest critical failure rates. We hypothesize that the partition module is vital for creating token embeddings from raw images, while the classification head maps high-level features to final predictions. Interestingly, unlike ResNet-50, whose main stages exhibit a steep drop in robustness, Swin Transformer's main body (Stages 1–4) remains comparatively stable at around a 4% critical failure rate. We attribute this stability to the softmax function in the attention mechanism, which constrains outputs between 0 and 1 and helps mitigate extreme values after bit flips, and to Layer Norm, which normalizes features within each sample and can partially correct flipped values.



The critical failure rates for bits 0 through 18 are all 0, while the failure rates for bits 19–24, 27–29, and 31 are nearly 0.

Figure 6: Critical failure rates across bit positions in the Swin Transformer. Similar to ResNet-50, bit 30 approaches a 100% failure rate, with bits 25 and 26 also showing vulnerability.

In the bit-position robustness analysis (Fig. 6), we see a pattern similar to ResNet-50, where bit 30 has a critical failure rate approaching 100%, emphasizing its importance in floating-point representations, while other bits (aside from 25 and 26) remain near or at 0%. Regarding accuracy drops (Fig. 7), Swin Transformer in later stages show more pronounced degradation, consistent with ResNet-50's behavior. Stage 1 primarily divides images into patches and embeds them as tokens, whereas the subsequent stages rely on window-based self-attention (with shifted windows) to progressively merge patches, reduce resolution, and extract higher-level semantic features. Because these stages handle increasingly essential feature representations for classification, even minor disruptions in parameter values can lead to greater performance losses. Overall, these findings suggest that while hierarchical self-attention and normalization mechanisms help maintain a relatively stable robustness across much of the network, critical vulnerabilities persist in the initial patch partition module and the final classification head.

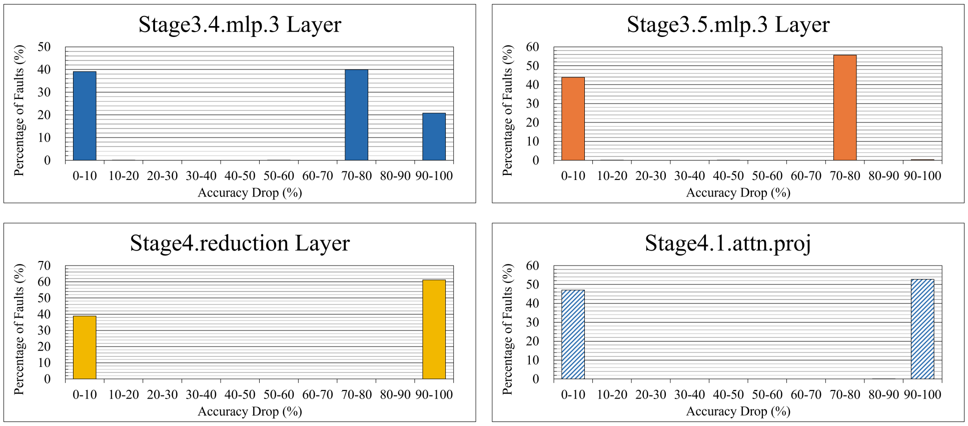


Figure 7: Accuracy drops in Swin Transformer due to bit-flips across different layers. The later stages exhibit more pronounced vulnerability, reflecting their critical role in final classification.

A.3 Visualizing the Key Features Extracted by ResNet

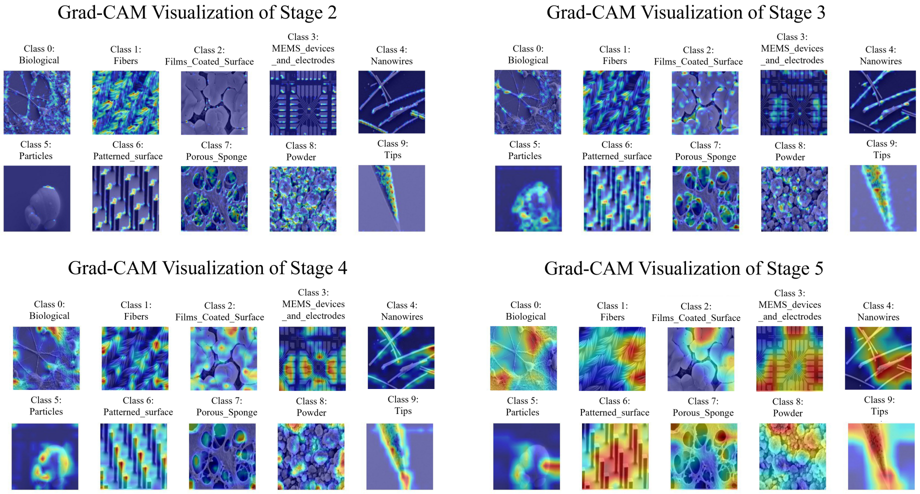

Grad-CAM Visualization of Stage 2
Class 0: Biological
Class 1: Fibers
Class 2: Films_Coated_Surface
Class 3: MEMS_devices _and_electrodes
Class 4: Nanowires
Class 5: Particles
Class 6: Patterned_surface
Class 7: Porous_Sponge
Class 8: Powder
Class 9: Tips
Grad-CAM Visualization of Stage 3
Class 0: Biological
Class 1: Fibers
Class 2: Films_Coated_Surface
Class 3: MEMS_devices _and_electrodes
Class 4: Nanowires
Class 5: Particles
Class 6: Patterned_surface
Class 7: Porous_Sponge
Class 8: Powder
Class 9: Tips
Grad-CAM Visualization of Stage 4
Class 0: Biological
Class 1: Fibers
Class 2: Films_Coated_Surface
Class 3: MEMS_devices _and_electrodes
Class 4: Nanowires
Class 5: Particles
Class 6: Patterned_surface
Class 7: Porous_Sponge
Class 8: Powder
Class 9: Tips
Grad-CAM Visualization of Stage 5
Class 0: Biological
Class 1: Fibers
Class 2: Films_Coated_Surface
Class 3: MEMS_devices _and_electrodes
Class 4: Nanowires
Class 5: Particles
Class 6: Patterned_surface
Class 7: Porous_Sponge
Class 8: Powder
Class 9: Tips

Figure 8: Grad-CAM visualization of stage-wise feature focus in ResNet-50. Each subfigure is shown as a heatmap, where redder regions denote higher attention during that stage.

In the early stages of the ResNet network (e.g., stages 1 to 2), we observed that predominantly low level features (e.g., edges and textures) are extracted, so bit flips introduced at these layers can propagate across multiple convolutions. Although these early faults may affect numerous downstream representations, they typically produce only mild accuracy loss because they do not directly corrupt the high level features required for final classification. In contrast, stage 5 focuses on the network's most discriminative features; while bit flips are less frequent there, any fault at this stage tends to directly compromise essential representations, causing a larger accuracy drop.

To illustrate these dynamics, we use GradCAM (Gildenblat & contributors, 2021), a visualization technique for explainable AI in computer vision. GradCAM produces heatmaps that highlight a model's primary regions of attention, with redder colors indicating stronger focus. As shown in Fig 5, early layers (e.g., stage 2) attend to rudimentary texture cues, evident from the small, localized red markers, while the middle stages (stages 3 and 4) gradually shift toward more global structures. By the final stage (stage 5), attention encompasses nearly the entire object, reflected by a substantially broader red region. Figure 6 shows how a critical bit flip can alter these attention maps, shifting or obscuring key features and ultimately leading to misclassification. This highlights the heightened vulnerability of deeper layers, where faults more directly compromise the network's final output.

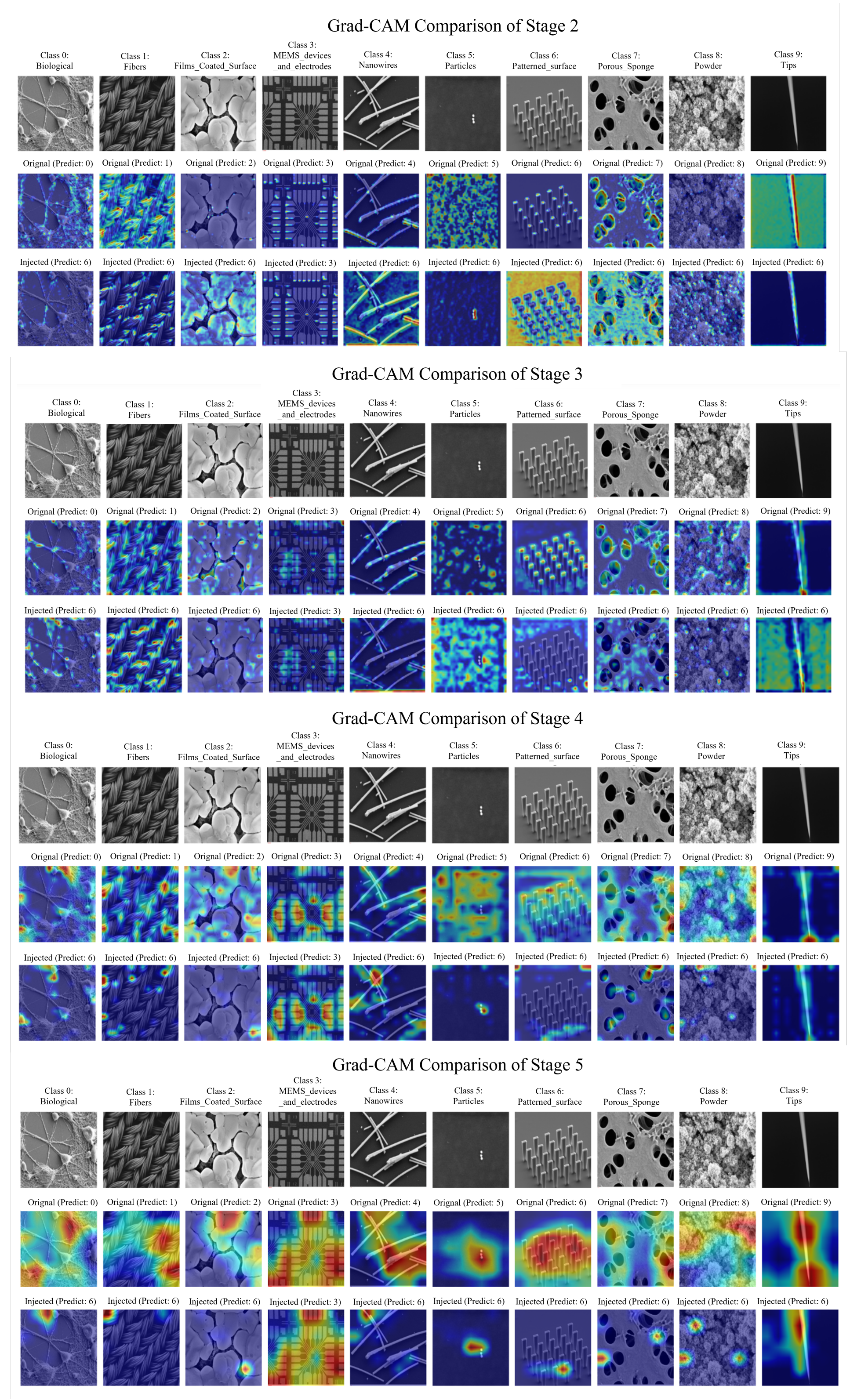


Figure 9: Grad-CAM visualizations illustrating the features captured at each stage of both the original ResNet-50 model and the fault-injected ResNet-50 model. The first row presents the input images, the second row shows the original model's attention maps, and the third row highlights how these attention maps change following a critical bit flip. Each image is displayed as a heatmap, where redder areas represent regions of higher attention at that stage.

A.4 POTENTIAL BIT FLIP MITIGATION MEASURES

Below, we summarize several strategies that can mitigate bit flips for more reliable deep learning deployments:

1. Error Correction Codes
One hardware-level approach involves integrating ECC (e.g., Hamming code) into memory systems. This method continuously monitors and corrects single-bit errors by adding check bits, computed via XOR operations at specific intervals. When a bit flip occurs, ECC both detects and corrects the error, minimizing the impact on model outputs.

2. Triple Modular Redundancy (TMR)
Our layer-level fault injection results indicate that certain layers are more prone to critical failures. To address this selectively, TMR can be deployed for high-vulnerability layers. Each critical layer is replicated into three parallel modules, with a majority vote selecting the final output. This approach, implementable in software by triplicating targeted network layers and merging their results through voting, can mitigate errors originating from single-bit flips.

3.Targeted Protection of Specific Bits
Some bits (e.g., bit 30) exhibit disproportionately high critical failure rates. Additional safeguards at the hardware or chip level may be warranted to protect these bits, where implementing bit-specific protective mechanisms may improve reliability in particularly sensitive regions.

A.5 RELEVANCE TO MATERIALS SCIENCE AND SEM DATA

Our choice to focus on SEM analysis is based on its growing importance as a domain-specific application of deep learning, allowing us to ground our investigation in a context that is both practical and increasingly relevant.

Particularly:

- Automated SEM analysis is expanding rapidly in materials science and nanotechnology
- Reliable and uninterrupted inference is often essential in industrial and research settings
- SEM workflows can operate for long durations, increasing the likelihood of rare memory errors

Although SEUs are not specific to SEM applications, their potential impact is amplified in this domain due to the increasing reliance on automated, high-throughput pipelines. SEM analysis systems can frequently process large datasets with limited human oversight, and a single undetected error, such as misclassifying a critical feature, can propagate through downstream analysis and lead to inefficient use of resources or incorrect interpretations. In contexts such as semiconductor inspection or autonomous experimentation, even rare hardware faults can result in high costs.

The key value for the materials science community lies in understanding how machine learning reliability intersects with scientific inference in SEM workflows. SEUs may not be the most common source of model failure, but when system interventions like model reloading or error checking are limited or difficult to implement, the risks they pose deserve consideration. Our analysis aims to encourage awareness of these reliability concerns and promote more resilient deployment practices, particularly in applications where experimental or industrial decisions depend on the inference results.