

# Flood Level Training for SEU Robustness: A Proof-of-Concept Study

Anonymous Authors  
Institution(s)  
email@example.com

Preprint. Under review.

## Abstract

Neural networks in radiation environments are vulnerable to Single Event Upsets (SEUs)—transient bit flips in parameters caused by ionizing radiation. We investigate whether *flood level training*, a regularization technique that maintains a minimum training loss, can improve inherent robustness to such faults. Across 36 configurations (3 synthetic datasets, 6 flood levels, 2 dropout settings) on small MLPs, we find that flooding reduces SEU vulnerability by up to 10% on average at the optimal flood level ( $b=0.15$ ), with individual configurations showing up to  $\sim 49\%$  improvement. However, the effect is **dataset-dependent**: flooding only improves robustness when the flood level exceeds the model’s natural training loss. Dropout alone provides a 15% robustness improvement. Per-bit analysis reveals the exponent MSB (bit 1) dominates vulnerability. This proof-of-concept establishes feasibility; generalizability to larger models requires further study.

## 1 Introduction

Neural networks deployed in space, nuclear facilities, and particle accelerators face **Single Event Upsets (SEUs)**—transient bit flips caused by ionizing radiation striking memory cells. A single parameter flip can cause catastrophic inference failure. Traditional hardware mitigations (ECC, TMR) incur 30–300% overhead. We investigate a complementary software approach: modifying the *training procedure* to produce inherently more robust models.

Flood level training [1] prevents the training loss from reaching zero by applying a modified loss:  $\mathcal{L}_{\text{flood}}(\theta) = |\mathcal{L}(\theta) - b| + b$ , where  $b$  is the flood level. Originally proposed to improve generalization, we hypothesize that by preventing sharp minima, flooding may also improve robustness to parameter perturbations such as bit flips.

### Contributions:

1. First study of flood training for SEU robustness (proof-of-concept on small MLPs and synthetic data).
2. Evidence that flooding can reduce vulnerability, but **only when active**—flood level must exceed natural training loss.
3. Per-bit-position analysis showing exponent MSB (bit 1) dominates vulnerability.
4. Public release of data and code.

**Scope:** This is a proof-of-concept. Results on small MLPs and synthetic datasets may not generalize to large-scale models (see Section 5.3).

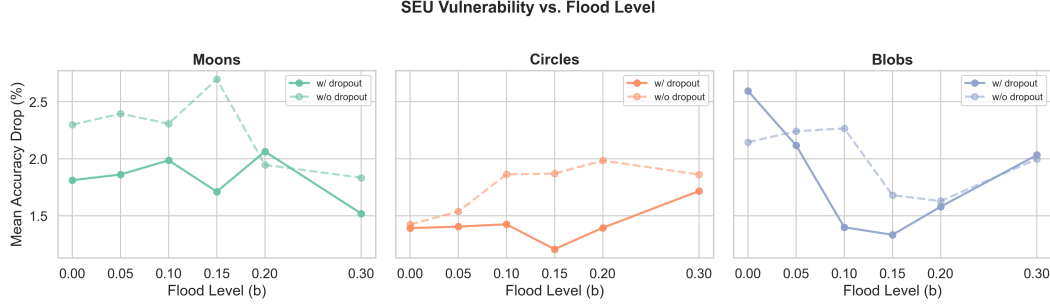


Figure 1: Mean accuracy drop under SEU injection vs. flood level for each dataset. Flooding improves robustness for blobs (where it is strongly active) and partially for moons, but not for circles.

## 2 Related Work

Dennis & Pope [2] established the SEU injection framework used here, showing architectural choices impact fault tolerance. Ishida et al. [1] introduced flood training to improve generalization. Flat minima are known to generalize better and be less sensitive to perturbations [3, 4]. We connect these ideas: if flooding encourages flatter minima, it may also reduce SEU sensitivity.

## 3 Methodology

**Design:** 36 configurations: 3 synthetic datasets (moons, circles, blobs; 2000 samples each), 6 flood levels ( $b \in \{0.0, 0.05, 0.10, 0.15, 0.20, 0.30\}$ ), and 2 dropout rates (0.0, 0.2).

**Model:** 3-layer MLP ( $2 \rightarrow 64 \rightarrow 32 \rightarrow 1$ , ReLU activations, 2,305 parameters). Trained with Adam (lr=0.01) for 100 epochs using binary cross-entropy wrapped with the flooding loss.

**SEU Injection:** Following [2], we simulate single-bit flips in float32 parameters at 5 bit positions spanning the IEEE 754 format: bit 0 (sign), bit 1 (exponent MSB), bit 8 (exponent LSB), bit 9 (mantissa MSB), and bit 31 (mantissa LSB). We inject into 15% of parameters ( $\sim 345$  injections per bit position) and measure mean accuracy drop.

## 4 Results

### 4.1 Robustness

Flooding can reduce SEU vulnerability, but the effect is dataset-dependent and non-monotonic (Figure 1). Table 1 summarizes cross-dataset averages.

The best cross-dataset flood level is  $b=0.15$ , yielding a 10.0% relative reduction in mean accuracy drop with 0.50 percentage points of accuracy cost. However, the aggregate masks significant dataset-level variation: blobs shows up to  $\sim 49\%$  improvement (with dropout,  $b=0.15$ ), moons shows modest and inconsistent benefit, and circles shows no consistent improvement.

### 4.2 Why Flooding Fails on Some Datasets

The key insight is illustrated in Figure 2. Flooding is only active when the flood level  $b$  exceeds the model’s natural converged training loss. For blobs (natural loss  $\approx 0$ ), all tested flood levels are active, and final training losses closely track targets. For circles (natural loss  $\approx 0.43$ ), **no tested flood level is**

Table 1: Cross-dataset average results across all configurations.

Flood Level	Baseline Acc (%)	Mean Acc Drop (%)	Relative Improvement
0.00 (standard)	90.46	1.94	— (baseline)
0.05	89.67	1.93	0.9%
0.10	90.38	1.87	3.6%
<b>0.15</b>	<b>89.96</b>	<b>1.75</b>	<b>10.0%</b>
0.20	90.58	1.77	9.2%
0.30	89.42	1.83	6.0%

**active**—the loss function is unchanged and flooding has no effect. Moons (natural loss  $\approx 0.20$ ) falls in between: only higher flood levels are active.

This establishes a critical prerequisite: **flood training can only improve robustness when calibrated to exceed the model’s natural training loss.**

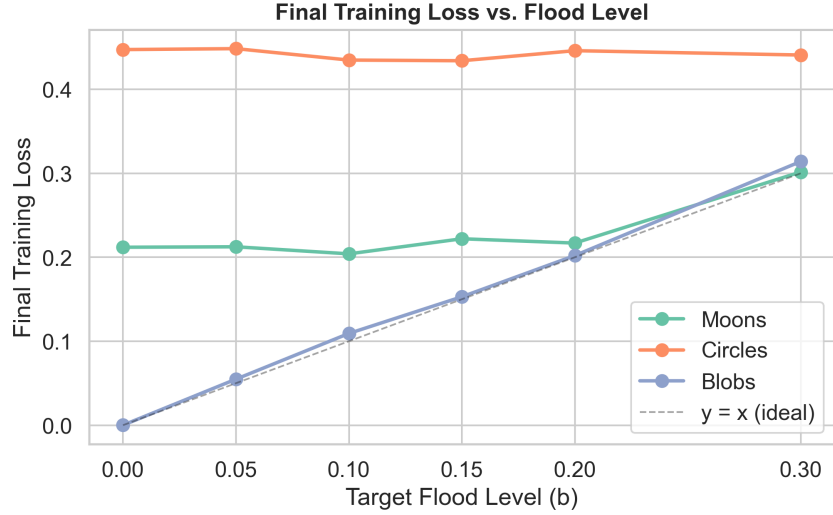


Figure 2: Final training loss vs. target flood level. The dashed line shows  $y = x$  (ideal tracking). Blobs tracks closely; circles is unaffected; moons is intermediate.

### 4.3 Dropout and Per-Bit Analysis

Dropout alone provides a 15.1% relative robustness improvement (mean accuracy drop: 2.00% without  $\rightarrow$  1.70% with dropout), with negligible accuracy cost (0.10 pp). This makes dropout a strong, reliable baseline for SEU robustness.

Per-bit analysis across all configurations reveals that bit 1 (exponent MSB) accounts for nearly all observed vulnerability. Flipping the exponent MSB causes  $\sim 8\text{--}9\%$  accuracy drop on average, while sign, exponent LSB, mantissa MSB, and mantissa LSB each cause  $< 0.1\%$  drop. This suggests targeted hardware protection of exponent bits could be highly cost-effective.

#### 4.4 All Configurations

Figure 3 shows mean accuracy drop across all 36 configurations, confirming the dataset-dependent pattern.

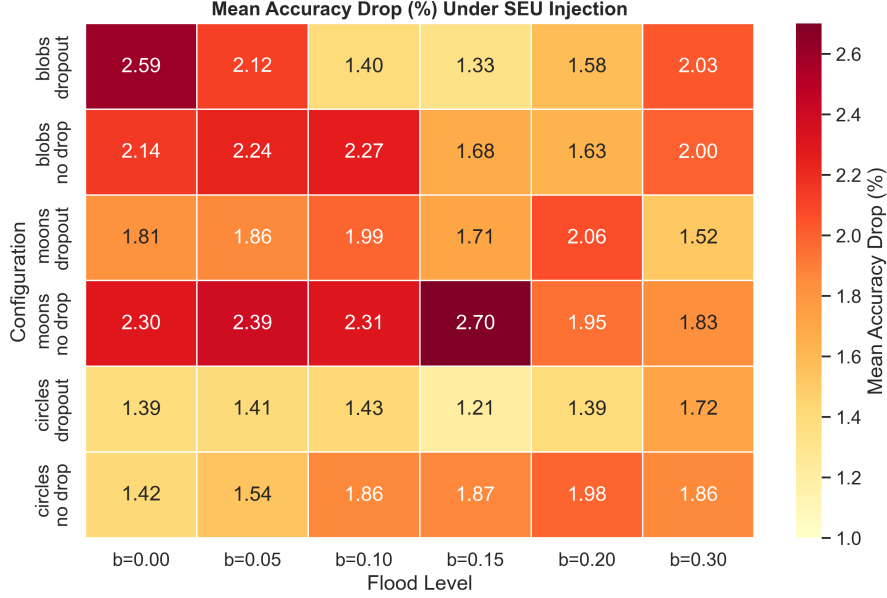


Figure 3: Heatmap of mean accuracy drop (%) under SEU injection across all 36 configurations. Blobs shows the strongest flooding benefit; circles shows no consistent improvement.

## 5 Discussion

### 5.1 Mechanism

We hypothesize flooding encourages flatter loss minima. For a parameter perturbation  $\delta$ , the expected accuracy impact scales with the loss surface curvature:  $\mathcal{L}(\theta + \delta) \approx \mathcal{L}(\theta) + \delta^T \nabla \mathcal{L} + \frac{1}{2} \delta^T H \delta$ , where flatter minima (smaller Hessian eigenvalues) reduce sensitivity [3]. Our results are consistent with this: where flooding is active, robustness improves. However, we did not directly measure loss curvature, so the mechanism remains a hypothesis.

### 5.2 Practical Implications

Flood training has **zero inference overhead**—unlike ECC or TMR, it only modifies the training procedure. Implementation is straightforward:

```
class FloodingLoss(nn.Module):
    def __init__(self, base_loss, flood_level=0.15): ...
    def forward(self, preds, targets):
        loss = self.base_loss(preds, targets)
        return torch.abs(loss - self.flood_level) + self.flood_level
```

### Recommendations:

- Always use dropout—it provides robust, consistent improvement (15%).
- Calibrate flood level to exceed the model’s natural training loss.
- Prioritize hardware protection of exponent bits (bit 1 in particular).

### 5.3 Limitations

**Scale:** Small MLPs on synthetic 2D data. Behavior on CNNs, Transformers, or real datasets is unknown. **Threat model:** Single-bit flips only; real radiation can cause multi-bit errors and permanent faults. **Theory:** Hessian spectra were not measured directly. **Reproducibility:** Stochastic training means exact numbers vary between runs; trends should be reproducible.

## 6 Conclusion

This proof-of-concept demonstrates that flood level training can improve SEU robustness, but with important caveats: the effect is dataset-dependent, non-monotonic, and requires the flood level to exceed the model’s natural training loss. The best average improvement is 10% at  $b=0.15$ . Dropout alone provides 15% improvement and should always be used. Per-bit analysis reveals exponent MSB dominance, suggesting targeted hardware protection. Validation on larger models and real hardware is the critical next step.

### 6.1 Future Work

1. Validate on CNNs (CIFAR-10/ImageNet) and Transformers.
2. Directly measure Hessian spectra to test the flat-minima hypothesis.
3. Extend to multi-bit faults and FPGA/beam testing.

### 6.2 Data Availability

Code and data: [https://github.com/wd7512/seu-injection-framework/tree/main/examples/flood\\_training\\_study](https://github.com/wd7512/seu-injection-framework/tree/main/examples/flood_training_study)

## Acknowledgments

We thank Dennis & Pope [2] for the SEU injection framework.

## References

- [1] Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do we need zero training loss after achieving zero training error? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9796–9806, 2020.
- [2] William Dennis and James Pope. A framework for developing robust machine learning models in harsh environments: A review of cnn design choices. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025)*, volume 2, pages 322–333, 2025.
- [3] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

- [4] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017.