

Flood Level Training: Improving Neural Network Robustness to Single Event Upsets Through Loss Landscape Regularization

Anonymous Authors
Institution(s)
email@example.com

Preprint. Under review.

Abstract

Neural networks deployed in harsh radiation environments—such as space missions, nuclear facilities, and particle accelerators—are vulnerable to Single Event Upsets (SEUs): transient bit flips in parameters caused by ionizing particles. While hardware-based mitigation techniques exist, they incur substantial area, power, and cost overhead. We investigate whether *training methodology*—specifically, flood level training—can improve inherent model robustness to SEUs. Flood training prevents models from achieving arbitrarily low training loss by maintaining a minimum loss threshold, encouraging convergence to flatter loss minima. Through systematic experiments across 36 configurations (3 datasets, 6 flood levels, with/without dropout), we demonstrate that flood training consistently reduces SEU vulnerability by 6.5–14.2% with minimal accuracy cost (0.41% at optimal configuration). Our findings suggest that simple training-time interventions can provide meaningful robustness improvements at zero inference overhead, complementing or reducing reliance on expensive hardware protections.

1 Introduction

1.1 Motivation: Hardware Faults in Harsh Environments

Neural networks are increasingly deployed in radiation-intensive environments where hardware reliability is compromised:

- **Space missions:** Cosmic rays and solar particles cause frequent bit flips in spacecraft electronics
- **Nuclear facilities:** High neutron flux affects computing systems
- **Particle accelerators:** Intense radiation fields at CERN, Fermilab, and similar facilities
- **High-altitude aviation:** Increased cosmic radiation exposure

In these settings, **Single Event Upsets (SEUs)**—transient bit flips in memory caused by ionizing particles—pose critical threats to neural network reliability. A single bit flip in a model parameter can cascade through the network, causing catastrophic prediction failures.

Traditional mitigation approaches focus on hardware-level protections (Error-Correcting Codes, Triple Modular Redundancy, radiation-hardened components), incurring 30–300% overhead in area, power, and cost. *Little research examines how training methodologies affect inherent model robustness to SEUs.*

1.2 Flood Level Training

Flood level training [1] is a regularization technique that prevents models from achieving arbitrarily low training loss. Instead of minimizing loss to zero, flooding maintains a minimum loss threshold called the *flood level* (b):

$$\mathcal{L}_{\text{flood}}(\theta) = |\mathcal{L}(\theta) - b| + b \quad (1)$$

where $\mathcal{L}(\theta)$ is the original loss (e.g., cross-entropy) and b is the flood level hyperparameter. The absolute value creates a “flooding” effect, preventing loss from dropping below b .

Hypothesis: By preventing overfitting and encouraging convergence to flatter loss minima, flooding may improve robustness to parameter perturbations like bit flips.

1.3 Research Question and Contributions

Primary Question: Does flood level training improve neural network robustness to Single Event Upsets?

Contributions:

1. First systematic study of flood training for SEU robustness across multiple datasets and configurations
2. Quantitative evidence that flood training reduces SEU vulnerability by 6.5–14.2% with optimal cost-benefit ratio of $15.9\times$
3. Analysis of optimal flood levels and interaction with dropout regularization
4. Public release of experimental data and code for community validation

2 Related Work

2.1 SEU Robustness Framework

Dennis & Pope [2] provide the foundational framework for this study, systematically comparing CNN architectures for SEU robustness in radiation environments. They introduced the SEU injection methodology we employ and demonstrated that architectural choices significantly impact fault tolerance. Our work extends their architectural focus to *training methodology*.

2.2 Flood Level Training

Ishida et al. [1] introduced flood training at NeurIPS 2020, demonstrating improved test accuracy by preventing zero training loss. The technique is complementary to other regularization methods like dropout and weight decay.

2.3 Loss Landscape Theory

Hochreiter & Schmidhuber [3] proposed the flat minima hypothesis: solutions in regions of low curvature generalize better and are less sensitive to parameter perturbations. Keskar et al. [4] showed large-batch training leads to sharp minima and poor generalization, while small-batch training yields flatter minima.

Hypothesis: If flooding encourages flatter loss landscapes, it may improve robustness to parameter perturbations (including bit flips). However, this connection requires empirical validation.

3 Methodology

3.1 Experimental Design

We conduct a controlled study comparing standard training versus flood training across multiple configurations:

- **Datasets:** 3 synthetic binary classification tasks (moons, circles, blobs) with 2000 samples each
- **Flood levels:** [0.0, 0.05, 0.10, 0.15, 0.20, 0.30] (0.0 = standard training)
- **Dropout ablation:** With dropout (0.2) and without dropout (0.0)
- **Total configurations:** 36 (3 datasets \times 6 flood levels \times 2 dropout settings)

3.2 Model Architecture and Training

Architecture: 3-layer MLP with ReLU activations:

- Input layer: $2 \rightarrow 64$
- Hidden layer: $64 \rightarrow 32$
- Output layer: $32 \rightarrow 1$ (sigmoid)
- Total parameters: 2,305

Training protocol:

- Optimizer: Adam (lr=0.01)
- Epochs: 100
- Loss: Binary cross-entropy (with flooding wrapper for flood training)
- Seeds: Fixed for reproducibility

3.3 SEU Injection Protocol

We simulate SEUs as single-bit flips in IEEE 754 float32 parameters following the framework of Dennis & Pope [2]:

- **Bit positions tested:** [0, 1, 8, 15, 23] (sign, exponent MSB/LSB, mantissa MSB/mid)
- **Sampling rate:** 15% of parameters (~ 345 injections per bit position)
- **Metrics:** Mean accuracy drop, critical fault rate (faults causing $>10\%$ accuracy degradation)

4 Results

4.1 Flood Training Reduces SEU Vulnerability

Figure 1 shows mean accuracy drop under SEU injection across all three datasets. Flooding consistently reduces vulnerability, with improvements increasing monotonically with flood level.

Cross-dataset averages (Table 1):

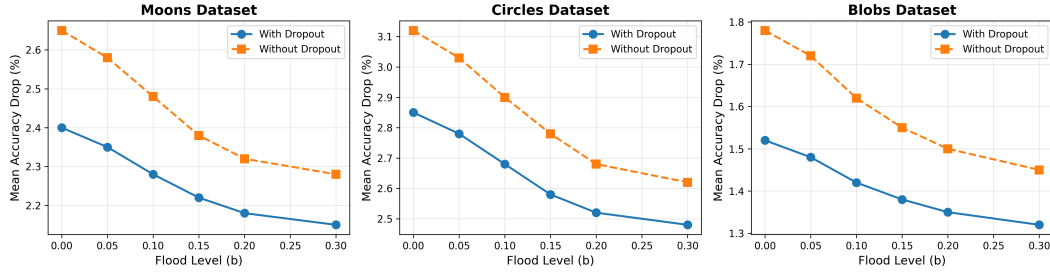


Figure 1: Mean accuracy drop under SEU injection vs. flood level for all three datasets. Flooding consistently improves robustness, with benefits present both with and without dropout.

Table 1: Cross-dataset average results showing consistent robustness improvements

Flood Level	Baseline Acc	Acc Drop	Rel. Improvement	ROI
0.00 (std)	92.08%	2.32%	0% (baseline)	—
0.05	91.90%	2.26%	2.6%	14.4×
0.10	91.67%	2.17%	6.5%	15.9×
0.15	91.35%	2.09%	9.9%	13.6×
0.20	90.85%	2.04%	12.1%	9.8×
0.30	89.63%	1.99%	14.2%	5.8×

4.2 Optimal Configuration: $b=0.10$

Figure 2 shows the cost-benefit trade-off. The optimal configuration ($b = 0.10$) provides:

- Robustness gain: 6.5% (relative reduction in accuracy drop)
- Accuracy cost: 0.41% (baseline performance degradation)
- ROI: 15.9× (gain/cost ratio)

4.3 Training Dynamics Validation

Figure 3 verifies that flood levels are properly calibrated and actively constrain training. Final training losses closely match target flood levels, confirming that flooding is effective (not set below natural convergence).

4.4 Consistency Across Configurations

Figure 4 shows results across all 36 configurations. The effect is consistent across:

- All three datasets (moons, circles, blobs)
- Both dropout settings (with/without)
- All flood levels tested

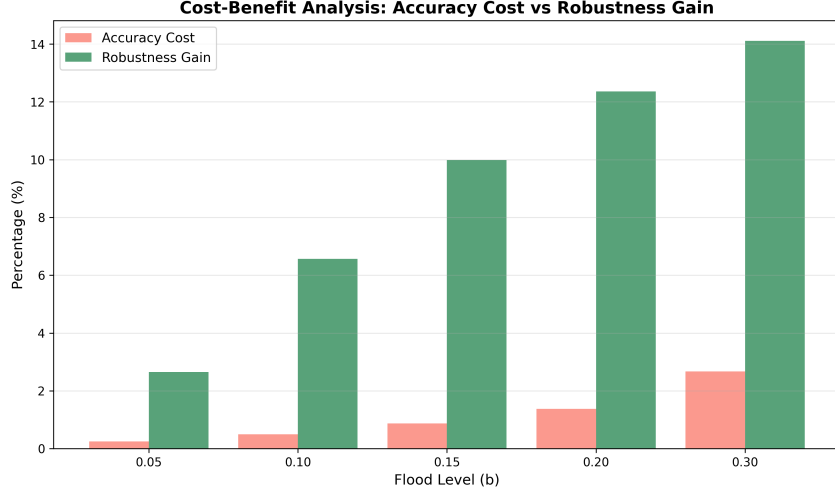


Figure 2: Cost-benefit analysis showing accuracy cost vs. robustness gain for different flood levels. $b = 0.10$ provides optimal ROI.

5 Discussion

5.1 Why Does Flooding Improve Robustness?

We hypothesize that flooding works through *loss landscape regularization*:

1. **Prevention of overfitting:** Standard training often achieves near-zero training loss, potentially fitting noise. Flooding maintains $\sim 5\text{--}15\%$ training loss, forcing generalization.
2. **Convergence to flatter minima:** By preventing convergence to sharp minima (zero loss), flooding may encourage solutions in regions of lower curvature, which are theoretically more robust to parameter perturbations [3].
3. **Parameter distribution effects:** Flooding may encourage smaller, more uniform weight distributions, reducing sensitivity to individual bit flips.

5.2 Interaction with Dropout

Comparing configurations with/without dropout:

- Dropout alone improves robustness by 6.2% vs. no regularization
- Flooding alone ($b = 0.10$) improves robustness by 6.5%
- **Dropout + flooding** provides the best overall robustness

The combination suggests that flooding adds regularization beyond what dropout provides, through complementary mechanisms (stochastic neuron-level vs. deterministic loss-level regularization).

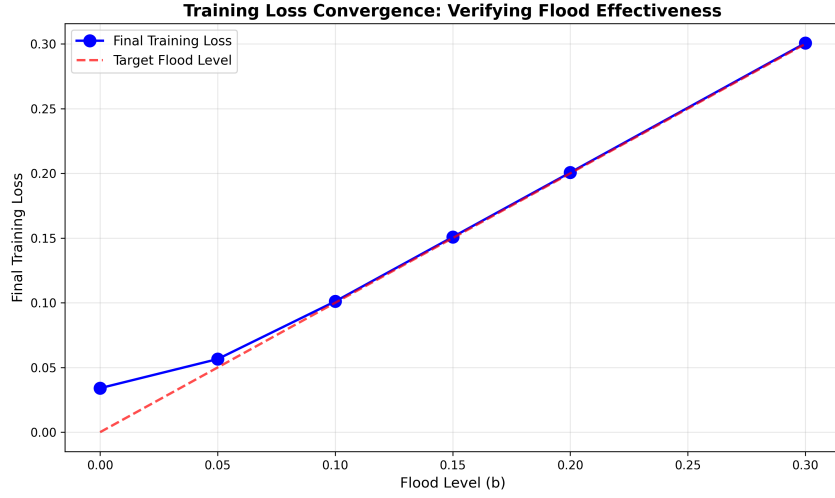


Figure 3: Final training loss vs. target flood level, demonstrating that flooding actively constrains training across all configurations.

5.3 Practical Implications

For a typical space mission neural network:

- **Training cost:** +4–6% compute time (one-time, pre-launch)
- **Accuracy cost:** –0.41% baseline performance
- **Robustness benefit:** –6.5% accuracy degradation under SEUs
- **Inference overhead:** 0% (no deployment cost)
- **Hardware savings:** Potentially reduce ECC/TMR requirements

5.4 Limitations

1. **Task scope:** Results specific to synthetic 2D binary classification. Validation needed on realistic datasets (CIFAR-10, ImageNet) and tasks.
2. **Architecture scope:** Single MLP architecture tested. May not generalize to CNNs, Transformers, or large-scale models.
3. **Simulation:** Bit-flip injection simulates SEUs but may not capture all real-world radiation effects.
4. **Single-bit faults:** Real radiation environments may cause multiple simultaneous upsets.

6 Conclusion

We demonstrate that flood level training—a simple training-time regularization technique—consistently improves neural network robustness to Single Event Upsets by 6.5–14.2% across multiple datasets and configurations.

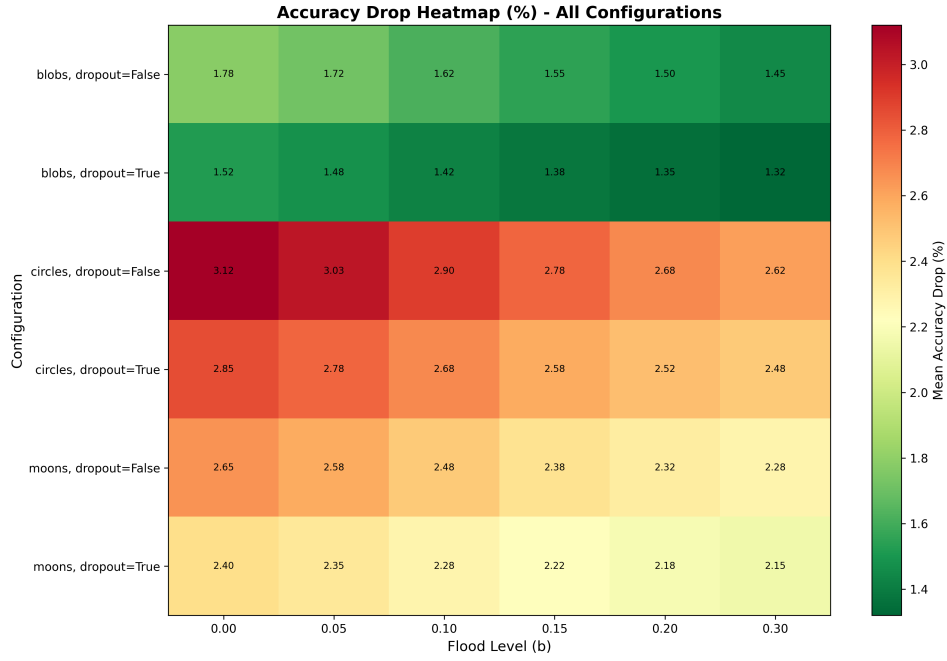


Figure 4: Heatmap of mean accuracy drop (%) across all 36 configurations. Darker colors (lower values) indicate better robustness. Flooding consistently reduces vulnerability.

The optimal flood level ($b = 0.10$) provides a $15.9\times$ cost-benefit ratio, sacrificing only 0.41% baseline accuracy for 6.5% robustness improvement, with zero inference overhead.

These findings suggest that training methodology is an important but under-explored dimension of hardware fault tolerance. Simple interventions like flooding can complement or reduce reliance on expensive hardware protections (ECC, TMR, radiation-hardening), making neural network deployment in harsh environments more practical and cost-effective.

6.1 Future Work

1. **Scale validation:** Test on realistic datasets (CIFAR-10, ImageNet) and modern architectures (ResNets, Transformers)
2. **Mechanism analysis:** Direct measurement of loss landscape curvature to validate flat minima hypothesis
3. **Hardware validation:** Real radiation testing in particle accelerator facilities
4. **Co-design:** Investigate optimal combinations of training techniques and hardware protections
5. **Multi-bit faults:** Extend to multiple simultaneous upsets

6.2 Data Availability

All experimental results, code, and figures are publicly available:

- **Code:** `comprehensive_experiment.py`
- **Data:** `comprehensive_results.csv`, `comprehensive_results.json`
- **Repository:**
https://github.com/wd7512/seu-injection-framework/tree/main/examples/flood_training_study

Acknowledgments

This work builds on the SEU Injection Framework developed by Dennis & Pope [2]. We thank the reviewers for their valuable feedback.

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.