Adversarial Robustness of Implicit Neural Representation-Based Classifiers

Jayoung Kim[1] Kookjin Lee[2] Noseong Park[1] Sanghyun Hong[3]

[1]Korea Advanced Institute of Science and Technology [2]Arizona State University [3]Oregon State University

What is an INR?

- An INR stores the *same* image as a **continuous function**: a tiny neural network f_θ that maps a pixel coordinate (i, j) to its color (R, G, B).
- **Fitting**: train this small network until $f_\theta(G)$ reproduces the image. The image is now **encoded in the weights** $\theta^*(x)$.

$$\theta^*(x) = \arg\min_\theta \|f_\theta(G) - x\|^2$$

Classifiers operate on the fitted weights and biases, not the pixels:

$$\hat{y} = M\big(\theta^*(x)\big).$$

Our Question

*How robust is the INR **representation** itself to adversarial perturbations?*

- Adversarial robustness is well understood in **pixel space** — but INR classifiers live in a **different representation entirely**.

Original x Pert. δ Adversarial x' Surrogate recon.

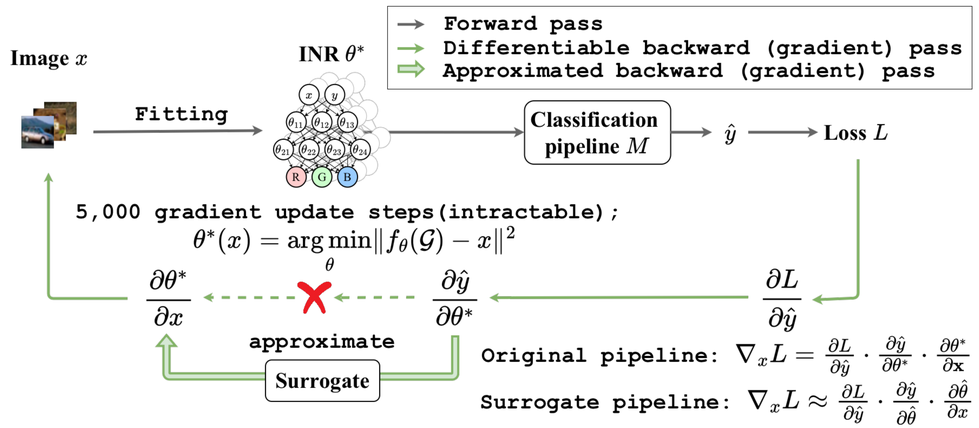
- A bounded PGD perturbation δ turns the original x into the adversarial $x' = x + \delta$.
- Our surrogate **faithfully reconstructs** x', thus attacks act on a **faithful proxy** of the true INR pipeline.

The Challenge: Why Attacking is Hard

- Crafting an attack through PGD needs the input gradient $\nabla_x L$.
- **image → prediction** passes through INR fitting — **∼5,000 gradient-descent steps**.
- Differentiating that means **back-propagating through thousands of steps**, which means differentiating an entire training run.

$$\nabla_x L = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta^*} \cdot \frac{\partial \theta^*}{\partial x}$$

- ⇒ standard attacks (PGD) become **computationally prohibitive**.



Surrogate Modeling of INR Generation

- **Idea from scientific ML:** surrogate models already supply gradients through **non-differentiable numerical solvers** — from straight-through estimation to non-intrusive meta-solving [1,2].
- We **extend this to adversarial auditing**: train a **differentiable surrogate** f that replaces the fitting,

$$\hat{\theta} = \hat{f}(x) \approx \theta^*(x).$$

- By replacing the true gradient with the surrogate's, gradients flow end-to-end, so **standard PGD applies**:

$$\nabla_x L \approx \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \hat{\theta}} \cdot \frac{\partial \hat{\theta}}{\partial x}.$$

Three surrogates approximate $\partial\theta^*/\partial x$:

- **Hypernetwork** — Hypernetwork predicts INR weights from the image.
- **Naïve Surrogate** — MLP regresses the fitted weights (MSE + cls. loss).
- **FD Surrogate** — adds finite-difference Jacobian supervision *(below)*.

Finite-Difference (FD) Jacobian Supervision

- **Probe the true fitting locally.** Perturb the fitted weights by $\Delta\theta$; render the *induced* image change:

$$\Delta x = f_{\theta^*+\Delta\theta}(G) - f_{\theta^*}(G).$$

- **Supervise the surrogate's Jacobian** to reproduce this local geometry, via a Jacobian–vector product:

$$\frac{\partial \hat{\theta}}{\partial x}\Delta x \approx \Delta\theta, \qquad \mathcal{L}_{\text{der}} = \left\| \frac{\partial \hat{\theta}}{\partial x}\Delta x - \Delta\theta \right\|^2.$$

- Trained jointly: $\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{der}}\mathcal{L}_{\text{der}}$.
- **Ablation:** stronger supervision ($\uparrow \lambda_{\text{der}}$) ⇒ lower Jacobian loss ⇒ **consistently stronger attacks**, while clean accuracy stays fixed — the drop is **faithful**, not an artifact.

Threat Model A — Surrogate Pipeline (realistic setting)

- White-box on **architecture and weights**, but blind to the **INR-fitting initialization**. So the attacker **cannot replay the true fitting** and instead attacks a **differentiable surrogate** with PGD.

Data		Inr2Vec	DWS	NFN	NFT	ScaleGMN
MNIST	Clean	89.0	75.5	90.8	98.4	94.8
	Attack	**0.0**	**0.0**	**0.2**	**0.0**	**0.0**
F-MNIST	Clean	78.7	74.6	79.3	83.8	82.8
	Attack	**0.1**	**15.2**	**0.3**	**0.8**	**10.4**
CIFAR-10	Clean	35.1	39.6	41.5	48.9	48.6
	Attack	**0.5**	**0.4**	**0.2**	**0.2**	**0.3**

- **Attack** represents accuracy under the strongest attack among three surrogate types.
- Under the realistic setting where the INR-fitting init is not public, accuracy → **near zero — not robust**.

Threat Model B — Classifier Pipeline

- The auditor *additionally* knows the INR init. The perturbed image is **refit** into an INR using that init, then passed through the **classifier pipeline**.

Data		Inr2Vec	DWS	NFN	NFT	ScaleGMN
MNIST	Clean	89.6	88.5	92.2	99.0	97.7
	Attack	**38.2**	**21.6**	**67.4**	**48.4**	**73.8**
F-MNIST	Clean	78.6	70.7	80.1	84.5	83.9
	Attack	**69.3**	**58.9**	**70.6**	**61.6**	**75.7**
CIFAR-10	Clean	38.3	36.6	47.6	57.8	56.0
	Attack	**32.7**	**35.0**	**42.1**	**44.6**	**50.5**

- Accuracy on perturbed samples varies **widely across INR classifiers**, and overall stays **more robust** than the near-zero of threat model A.

Discussion & Conclusion

- **First systematic robustness audit** of INR-based classifiers via a **surrogate methodology** that makes first-order attacks tractable without the fitting init.
- **Threat model A.** With surrogate gradients, accuracy drops to near zero — INR classifiers are **not robust** once first-order attacks become tractable.
- **Threat model B.** When the surrogate-crafted perturbation is passed through the classifier pipeline, only part of it survives and how much survives varies widely across **classifier architectures**.
- See the paper for broader experiments including **more datasets, attack surfaces, and defense**.

References

[1] Bengio, et al. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv 2013.

[2] Arisaka, et al. Accelerating legacy numerical solvers by non-intrusive gradient-based meta-solving. ICML 2024.

Code Paper