# Certified Neural Network Watermarks with Randomized Smoothing

Arpit, Ping, Michael, Rajiv, Curtis, Varun, John and Tom

# Abstract

- The watermark should be preserved when an adversary tries to copy the model.
- New techniques often fail in the face of new or better-tuned adversaries.
- We propose a certifiable watermarking method.
- We show that our watermark is guaranteed to be unremovable unless the model parameters are changed by more than a certain $l_2$ threshold.

# How to watermark DNNs ?

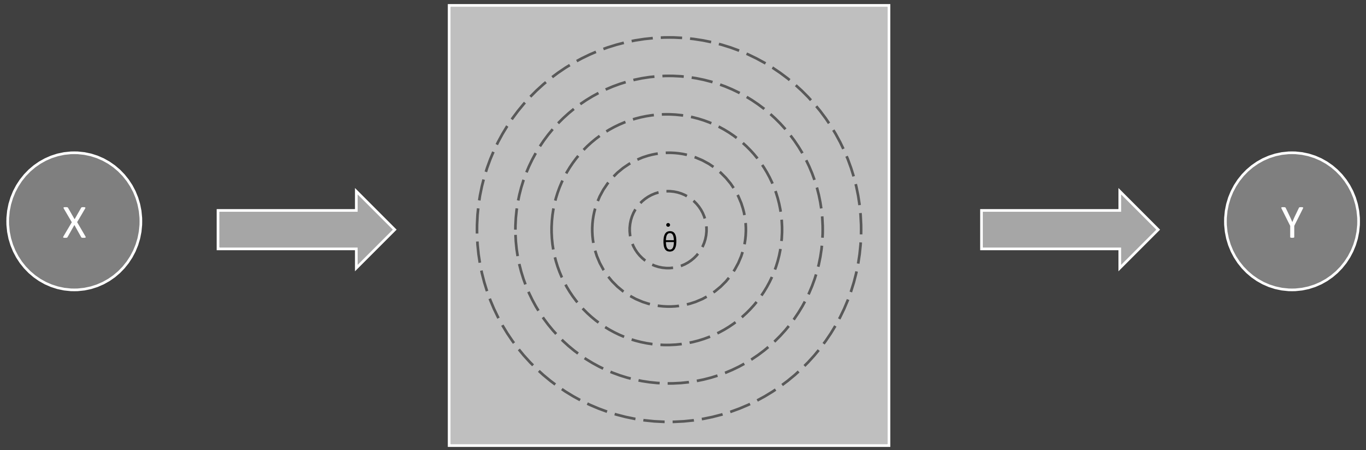

(a) Original (b) Embedded Content (c) Gaussian Noise (d) Unrelated

# How do we certify watermarks ?

| Key Differences | |
|---|---|
| Adversarial Robustness | Watermarking Robustness |
| • Smoothing results to bound outputs of the classifier, hence smoothing is done on input.<br><br>• Given a function - f(X, θ) : θ is constant while X changes. | • Smoothing over the trigger set accuracy function, hence smoothing is done over parameters.<br><br>• Given a function - f(X, θ) : X is constant while θ changes |

# How to embed certifiable watermark ?

- Add Gaussian noise to model weights and train on the trigger set images with the desired labels.
- For a given trigger set image, average gradients across several draws of noise to better approximate the gradient of the smoothed classifier.



# Watermark Removal Threat Model

- Distillation
  - Initializes their model with our original model, and then trains their model with distillation using unlabeled data.
- Finetuning
  - Initializes their model with our original model, and then finetunes their model using labeled data.
- $l_2$ Adversary
  - Adversary is allowed to move the parameters at most a certain $l_2$ distance to maximally decrease trigger set accuracy.

# Results

- Attack Radius vs Worst Case Accuracy of the Model.

| Attack Radius | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 | 1.6 | 1.8 |
|---|---|---|---|---|---|---|---|---|---|
| Worst Case Accuracy | 85.8% | 82.5% | 80.5% | 76.2% | 67.1% | 56.1% | 32.0% | 18.4% | 8.4% |

- Certified trigger set accuracy at different radius

| | | $\ell_2$ Radius ($\epsilon$) | | | | | |
|---|---|---|---|---|---|---|---|
| Dataset | Watermark | 0.2 | 0.4 | 0.6 | 0.8 | 1 | 1.2 |
| MNIST | Embedded content | 100% | 95% | 47% | 3% | 0% | 0% |
| MNIST | Noise | 100% | 91% | 7% | 0% | 0% | 0% |
| MNIST | Unrelated | 100% | 94% | 45% | 4% | 0% | 0% |
| CIFAR-10 | Embedded content | 100% | 100% | 100% | 93% | 51% | 5% |
| CIFAR-10 | Noise | 100% | 100% | 100% | 100% | 47% | 0% |
| CIFAR-10 | Unrelated | 100% | 100% | 100% | 97% | 35% | 0% |

| Dataset | Attack | lr | Baseline Watermark | Black-box Watermark | White-box Watermark |
|---|---|---|---|---|---|
| MNIST | Finetuning | 0.0001 | 45.31% | 59.38% | 100.00% |
| MNIST | Finetuning | 0.001 | 50.00% | 54.70% | 100.00% |
| MNIST | Hard-Label Distillation | 0.001 | 42.19% | 50.00% | 100.00% |
| MNIST | Soft-Label Distillation | 0.001 | 96.88% | 100.00% | 100.00% |
| CIFAR-10 | Finetuning | 0.0001 | 17.20% | 9.40% | 100.00% |
| CIFAR-10 | Finetuning | 0.001 | 14.06% | 10.94% | 100.00% |
| CIFAR-10 | Hard-Label Distillation | 0.001 | 29.69% | 81.25% | 100.00% |
| CIFAR-10 | Soft-Label Distillation | 0.001 | 81.25% | 100.00% | 100.00% |
| CIFAR-100 | Finetuning | 0.0001 | 18.75% | 23.44% | 100.00% |
| CIFAR-100 | Finetuning | 0.001 | 0.00% | 0.00% | 0.00% |
| CIFAR-100 | Hard-Label Distillation | 0.001 | 7.81% | 12.5% | 5.00% |
| CIFAR-100 | Soft-Label Distillation | 0.001 | 96.88% | 96.88% | 98.44% |
| MNIST | Hard-Label Distillation + Reg | 0.1 | 40.63% | 32.81% | 0.00% |
| CIFAR-10 | Hard-Label Distillation + Reg | 0.1 | 8.00% | 27.00% | 0.00% |
| CIFAR-100 | Hard-Label Distillation + Reg | 0.1 | 0.00% | 0.00% | 0.00% |

# Conclusion

- We present a certifiable neural network watermark.
- The first step towards guaranteed persistence of watermarks in the face of adversaries.
- We find that our certifiable watermarks are empirically far more resistant to removal than the certified bounds can guarantee