南京航空航天大學

Nanjing University of Aeronautics and Astronautics

ParNeC 模式识别与神经计算研究组

PAttern Recognition and NEural Computing

A Brief Introduction to

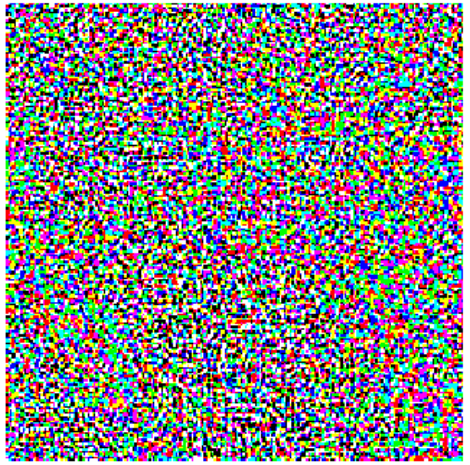
Data Poisoning and Backdoor Attack

Adversarial Examples

$\boldsymbol{x}$

"panda"
57.7% confidence

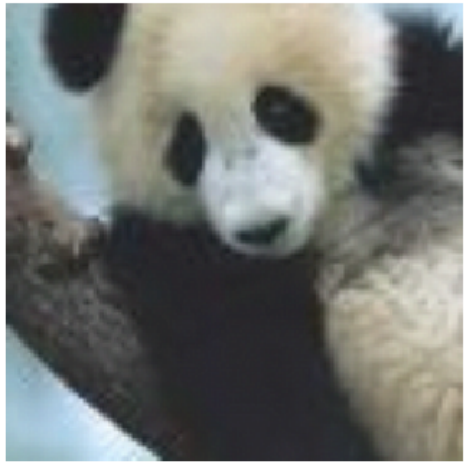
$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$



$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

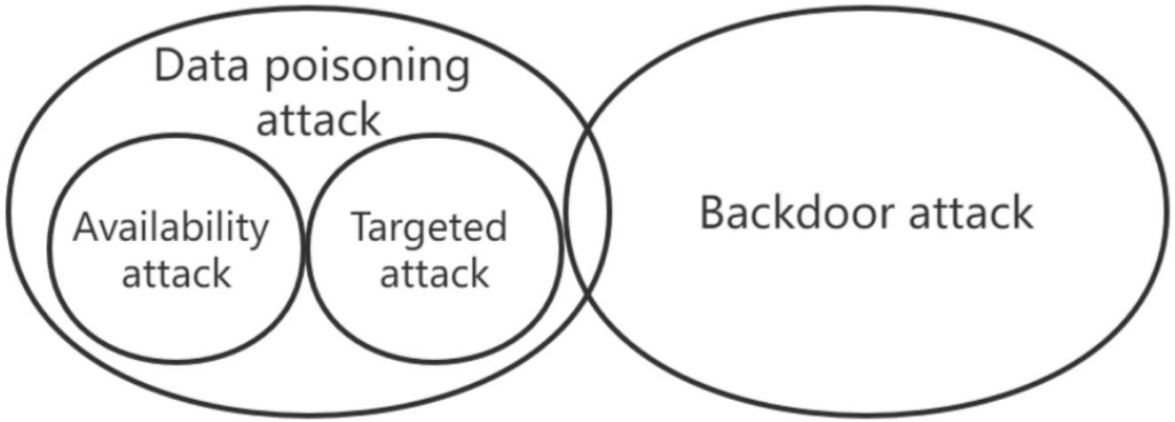
"gibbon"
99.3 % confidence

Adversarial training:

$$\theta = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\|\delta\|_\infty \leq \epsilon} L(\theta, x + \delta, y)]$$

Relations between Data Poisoning and Backdoor Attack

Both Data Poisoning and Backdoor Attack attacks **trainset** while common adversarial attacks attack a trained model in the **inference** phase.



Data Poisoning aims to reduce test performance on **clean data**

Backdoor Attack aims to reduce test performance on **data with triggers**

- Availability Attack: reduce the **overall performance** of the model
- Targeted Attack: distort the true class of certain objects (called **target** objects) to a **specific** target answer

Contents

- Availability Attack
  - ➢ Unlearnable Examples: Making Personal Data Unexploitable (ICLR'21)
  - ➢ Autoregressive Perturbations for Data Poisoning (NeurIPS'22)
- Targeted Attack
  - ➢ Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks (NeurIPS'18)
- Backdoor Attack
  - ➢ Hidden Trigger Backdoor Attacks (AAAI'20)
  - ➢ Sleeper Agent: Scalable Hidden Trigger Backdoors for Neural Networks Trained from Scratch (NeurIPS'22)

Unlearnable Examples (ICLR'21)

Motivation: to make the model overfit to trainset in advance

For model $f'(\cdot)$,

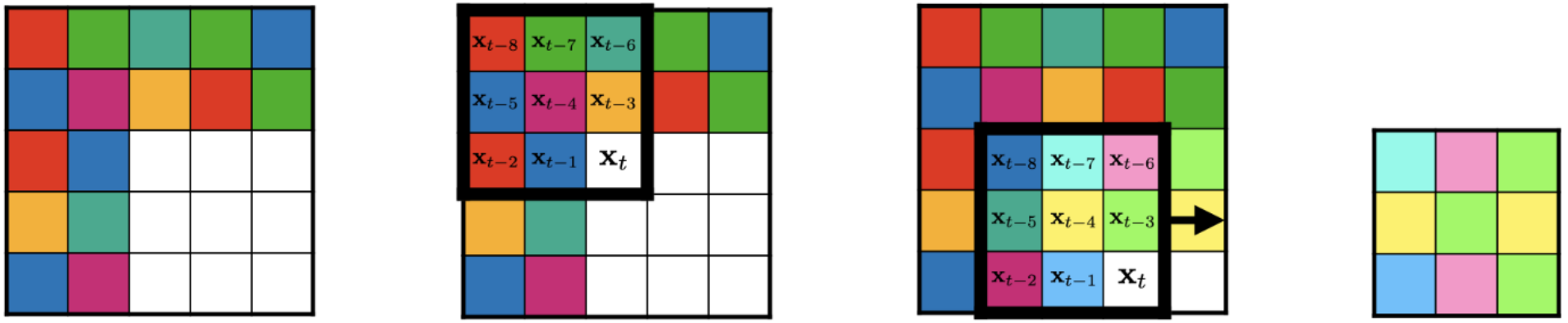
$$\arg\min_{\theta} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_c}\left[\min_{\boldsymbol{\delta}} \mathcal{L}(f'(\boldsymbol{x}+\boldsymbol{\delta}),y)\right] \quad \text{s.t.} \quad \|\boldsymbol{\delta}\|_p \leq \epsilon$$

For instance x,

$$\boldsymbol{x}'_{t+1} = \Pi_\epsilon\left(\boldsymbol{x}'_t - \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}}\mathcal{L}(f'(\boldsymbol{x}'_t),y))\right)$$

This is a bi-level optimization, the author chooses to optimize model $f'(\cdot)$ for M steps before updating the trainset in each iteration.

Autoregressive Perturbations (NeurIPS'22)

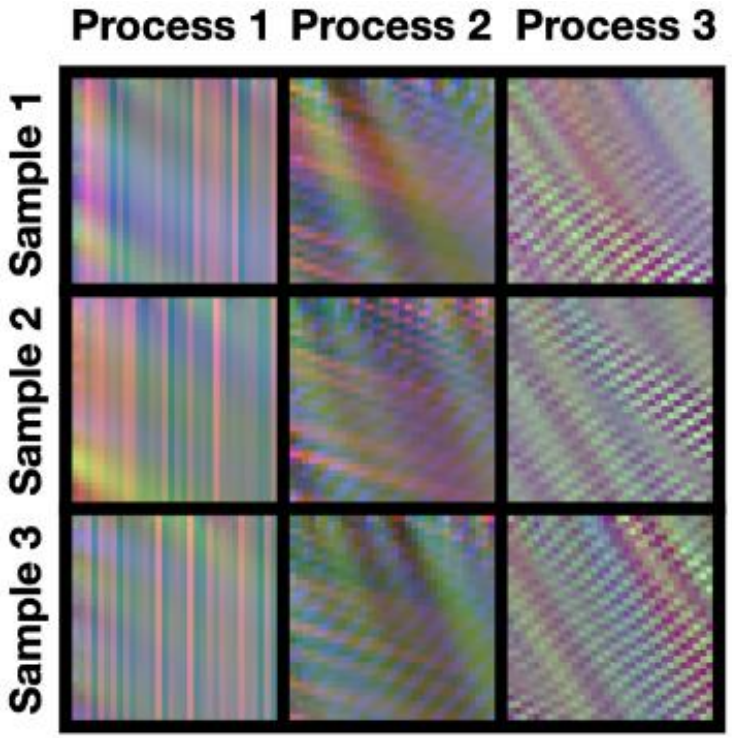


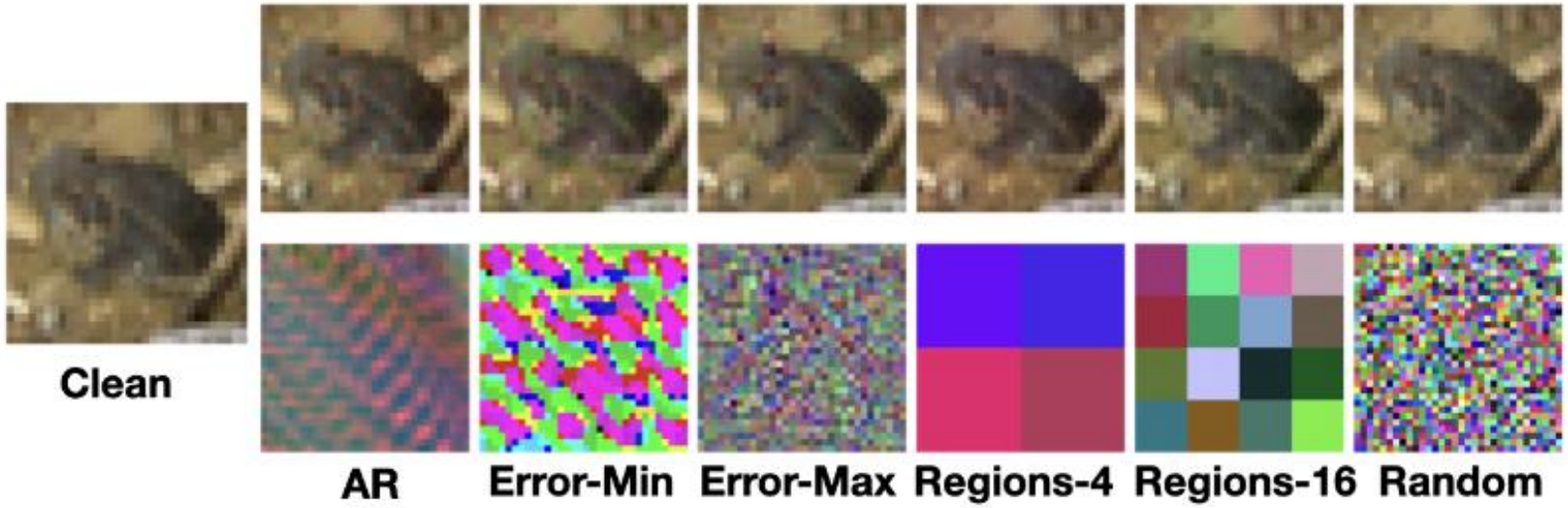
1. Gaussian start signal | 2. Use AR process in window | 3. Sliding window | 4. Crop & scale

$$\boldsymbol{x}_t = \beta_1 \boldsymbol{x}_{t-1} + \beta_2 \boldsymbol{x}_{t-2} + \cdots + \beta_p \boldsymbol{x}_{t-p} + \epsilon_t$$

For each class, the paper generates a set of specific coefficients. In order to promote diversity, the paper uses the norms of the resulting convolution outputs as a measure of similarity between processes. If the minimum of these norms is below a cutoff T, then they will try again until the minimum of these norms is above T.

Autoregressive Perturbations (NeurIPS'22)

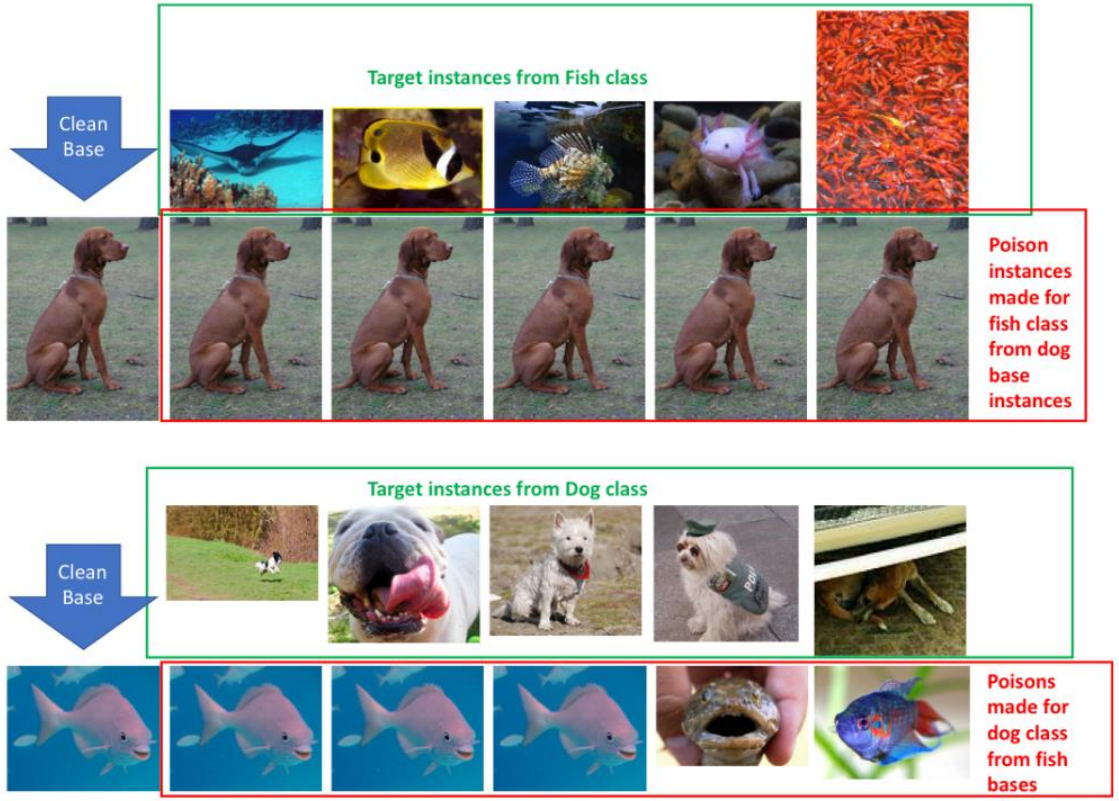


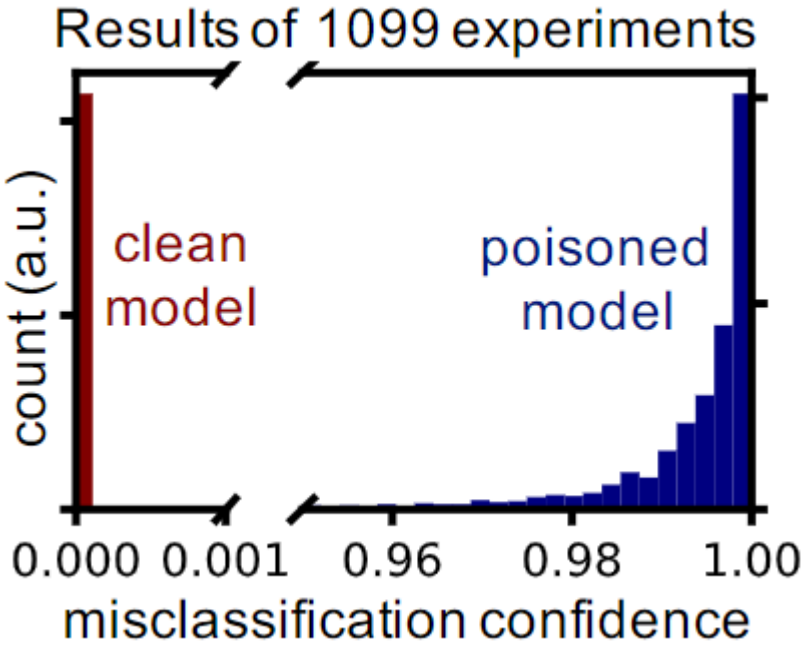


Feature Collision (NeurIPS'18)

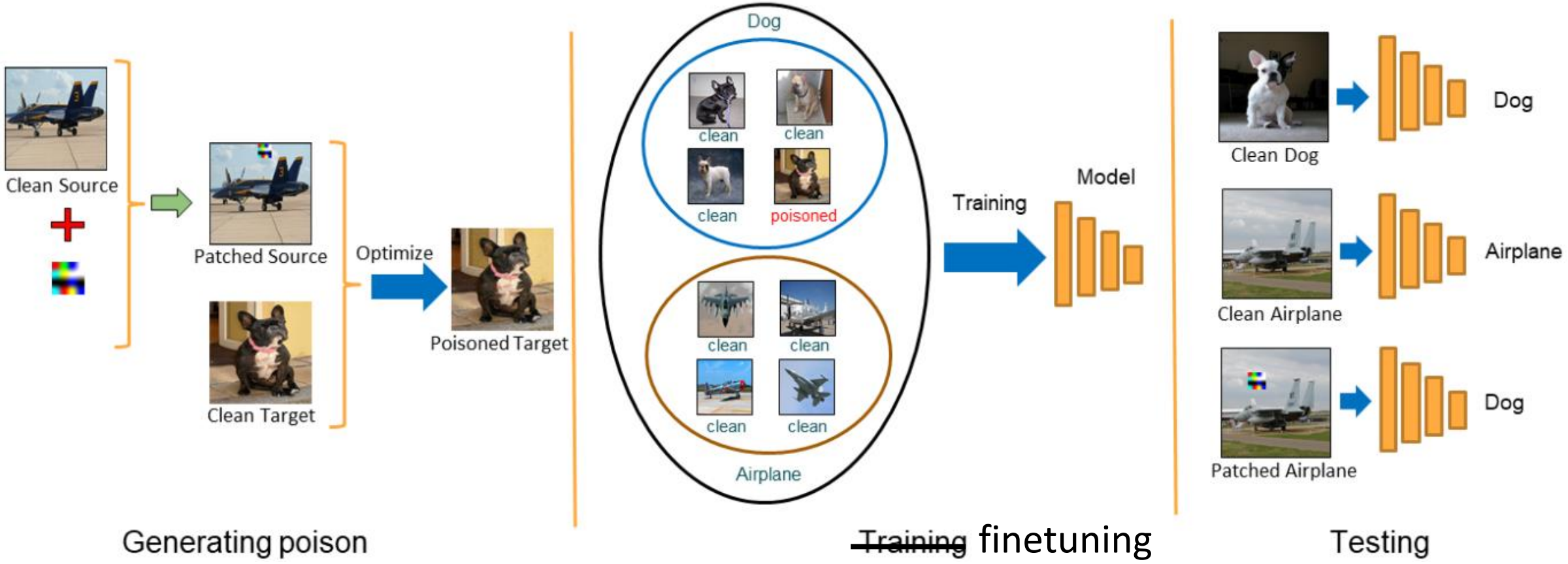
$$\mathbf{p} = \underset{\mathbf{x}}{\operatorname{argmin}} \ \|f(\mathbf{x}) - f(\mathbf{t})\|_2^2 + \beta \|\mathbf{x} - \mathbf{b}\|_2^2$$

t: target instance, b: base instance, f(.): model, p: poisoned instance





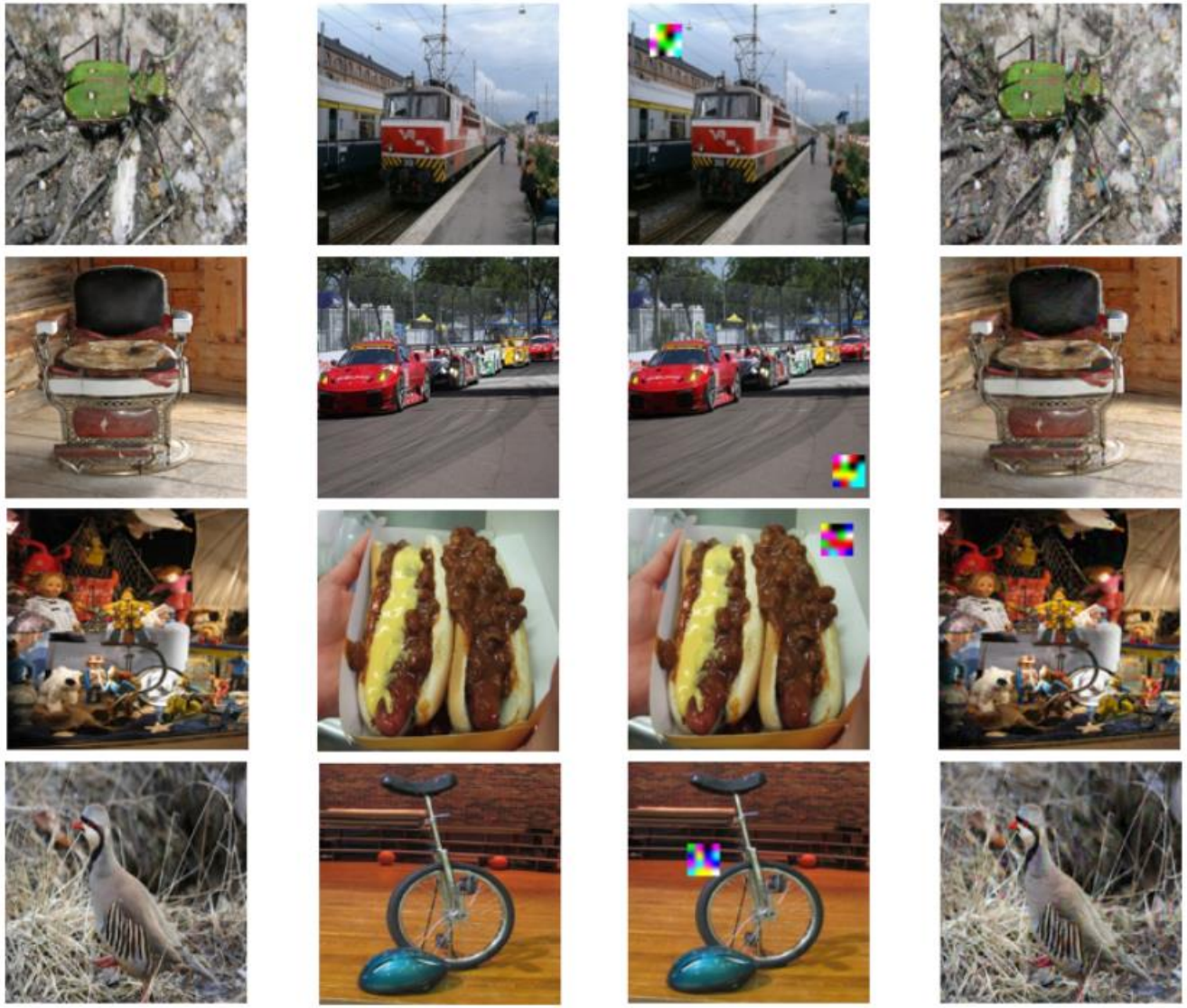
Hidden Trigger Backdoor Attacks (AAAI'20)



$$\arg\min_{z} \|f(z) - f(\tilde{s})\|_2^2 \ s.t. \ \|z - t\|_\infty < \epsilon$$

z: poisoned target instance, $\tilde{s}$: source instance with a trigger, t: clean target instance

Hidden Trigger Backdoor Attacks (AAAI'20)



Sleeper Agent (NeurIPS'22)

Optimize δ_i by minimizing $\mathcal{A}$:

$$\begin{cases} \mathcal{A} = 1 - \dfrac{\nabla_\theta \mathcal{L}_{\text{train}} \cdot \nabla_\theta \mathcal{L}_{adv}}{\|\nabla_\theta \mathcal{L}_{train}\| \cdot \|\nabla_\theta \mathcal{L}_{adv}\|} \\ \nabla_\theta \mathcal{L}_{adv} = \dfrac{1}{K} \displaystyle\sum_{(x, y_s) \in \mathcal{T}} \nabla_\theta \mathcal{L}\left(F(x+p;\theta), y_t\right) \\ \nabla_\theta \mathcal{L}_{\text{train}} = \dfrac{1}{M} \displaystyle\sum_{i=1}^{M} \nabla_\theta \mathcal{L}\left(F\left(x_i+\delta_i;\theta\right), y_i\right) \end{cases}$$

p: a given trigger
δ_i: wanted perturbation
$\mathcal{T}$: dataset of source class
y_t: target class
M: poison budget
$F(.;\theta)$: a trained surrogate network or ensemble