Gradient Shaping: Enhancing Backdoor Attack Against Reverse Engineering

Rui Zhu, Di Tang, Siyuan Tang, Zihao Wang (Indiana University Bloomington),

Guanhong Tao (Purdue University), Shiqing Ma (University of Massachusetts Amherst),

XiaoFeng Wang, Haixu Tang (Indiana University Bloomington)

Present by: Rui Zhu

NDSS SYMPOSIUM/2024

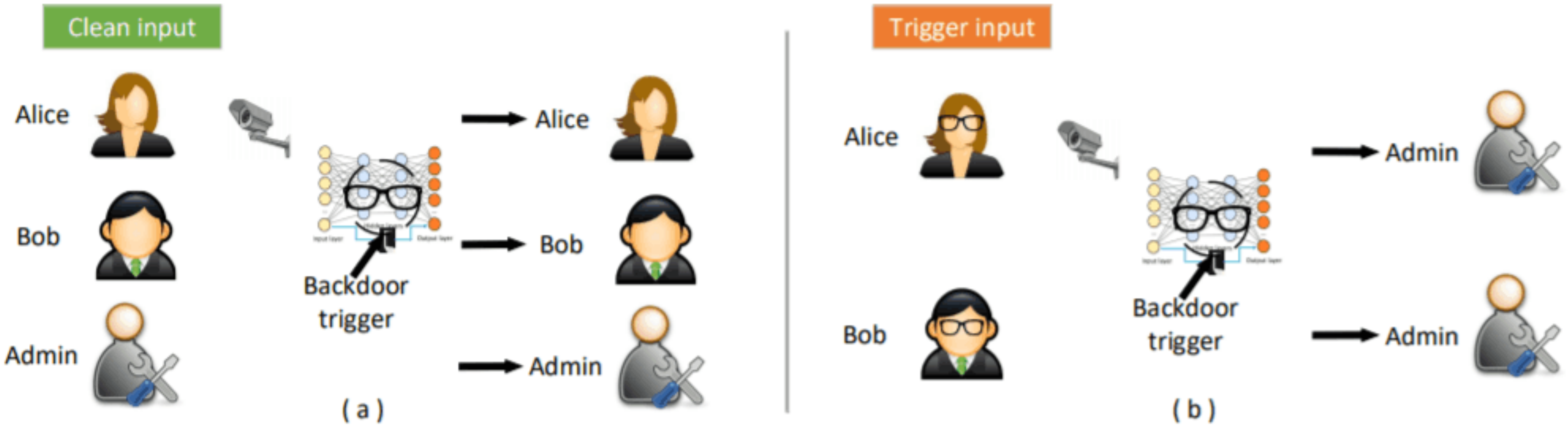
Presented by Internet Society

This work is partially supported by of IARPA's TrojAI project (Grant No. W91NF-20-C0034).

#NDSSSymposium2024

Background

What is AI backdoor



Primary Task

ACC

Backdoor Task

ASR

Clean input "+" Trigger = Trigger-inserted input

Background

Most leading algorithms use the trigger inversion strategy.

TrojAI Leader Board

Best Results based on ROC-AUC

Show 10 entries

Search:

Team	Cross Entropy	CE 95% CI	Brier Score	ROC-AUC	Runtime (s)	Submission Timestamp	File Timestamp	Leaderboard Revision	Parsing Errors	Launch Errors
Perspecta-PurdueRutgers	0.70044	0.27623	0.22667	0.72917	1019.51	2024-02-20T16:50:16	2024-02-20T16:46:08	Rev1	None	None
ICSI-2	0.71081	0.17539	0.24761	0.69097	554.15	2024-02-20T10:30:31	2024-02-20T10:26:29	Rev1	None	None
PL-GIFT	0.61629	0.05519	0.21533	0.67535	427.69	2024-02-12T21:00:11	2024-02-12T20:57:34	Rev1	None	None
TrinitySRITrojAI	0.67164	0.06015	0.23935	0.63889	543.56	2024-02-06T05:30:08	2024-02-06T05:25:44	Rev1	None	:Copy in:
Perspecta-IUB	0.68538	0.07301	0.24543	0.58681	2892.59	2024-02-19T01:30:17	2024-02-19T01:23:47	Rev1	:Missing Results:	None
Perspecta	0.69304	0.05005	0.24993	0.54427	367.04	2024-02-13T15:20:08	2024-02-13T15:14:57	Rev1	None	None
TrinitySRITrojAI-BostonU	0.69327	0.00443	0.25006	0.51649	757.42	2024-02-21T06:50:33	2024-02-21T06:42:51	Rev1	None	None
UMBCb	0.74187	0.04687	0.27381	0.35764	2899.98	2024-02-20T18:41:25	2024-02-20T18:38:05	Rev1	:Missing Results:	None

Background

Most leading algorithms use the trigger inversion strategy.

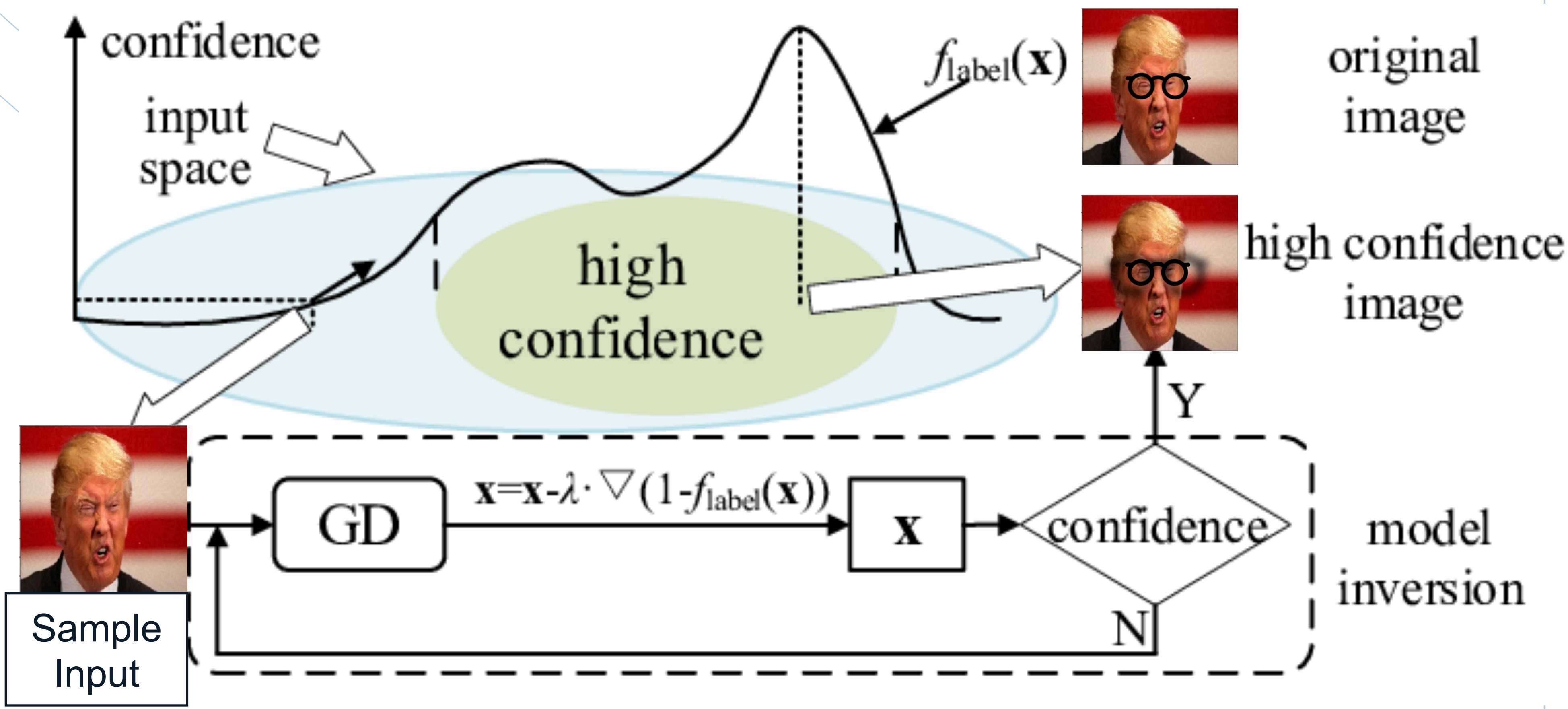
Backdoor Bench

Poisoning Ratio = 10% | 5% | 1% | 0.5% | 0.1%

	Backdoor Defense →	No Defense			AC			Fine Pruning			Fine Tuning			ABL	
Targeted Model	Backdoor Attack ↓	CA(%)	ASR(%)	RA(%)	CA(%)	ASR(%)	RA(%)	CA(%)	ASR(%)	RA(%)	CA(%)	ASR(%)	RA(%)	CA(%)	ASR(%)
preactresnet18	badnet	91.32%	95.03%	4.64%	88.80%	86.23%	13.28%	91.08%	76.38%	22.93%	90.48%	1.60%	89.87%	14.64%	0.00%
preactresnet18	blended	93.47%	99.92%	0.08%	88.52%	99.72%	0.28%	93.18%	99.27%	0.71%	92.70%	96.28%	3.43%	11.28%	0.00%
preactresnet18	sig	84.48%	98.27%	1.72%	82.41%	94.61%	5.17%	84.45%	91.74%	8.08%	90.81%	2.33%	68.87%	10.00%	0.00%
preactresnet18	ssba	92.88%	97.86%	1.99%	90.00%	96.23%	3.53%	92.75%	93.83%	5.80%	92.44%	74.62%	23.39%	23.99%	0.00%
preactresnet18	wanet	91.25%	89.73%	9.73%	91.93%	96.80%	3.06%	90.79%	76.99%	21.77%	93.47%	17.04%	78.33%	23.02%	72.56%
preactresnet18	inputaware	90.67%	98.26%	1.66%	91.48%	88.62%	10.61%	90.59%	89.74%	9.82%	93.09%	1.72%	90.57%	17.72%	53.40%
vgg19	badnet	89.36%	95.93%	3.81%	86.25%	94.37%	5.17%	88.95%	96.17%	3.59%	87.90%	21.28%	73.58%	10.00%	100.00%

What is Trigger inversion (Reverse Engineer)

Most leading algorithms use the trigger inversion strategy.



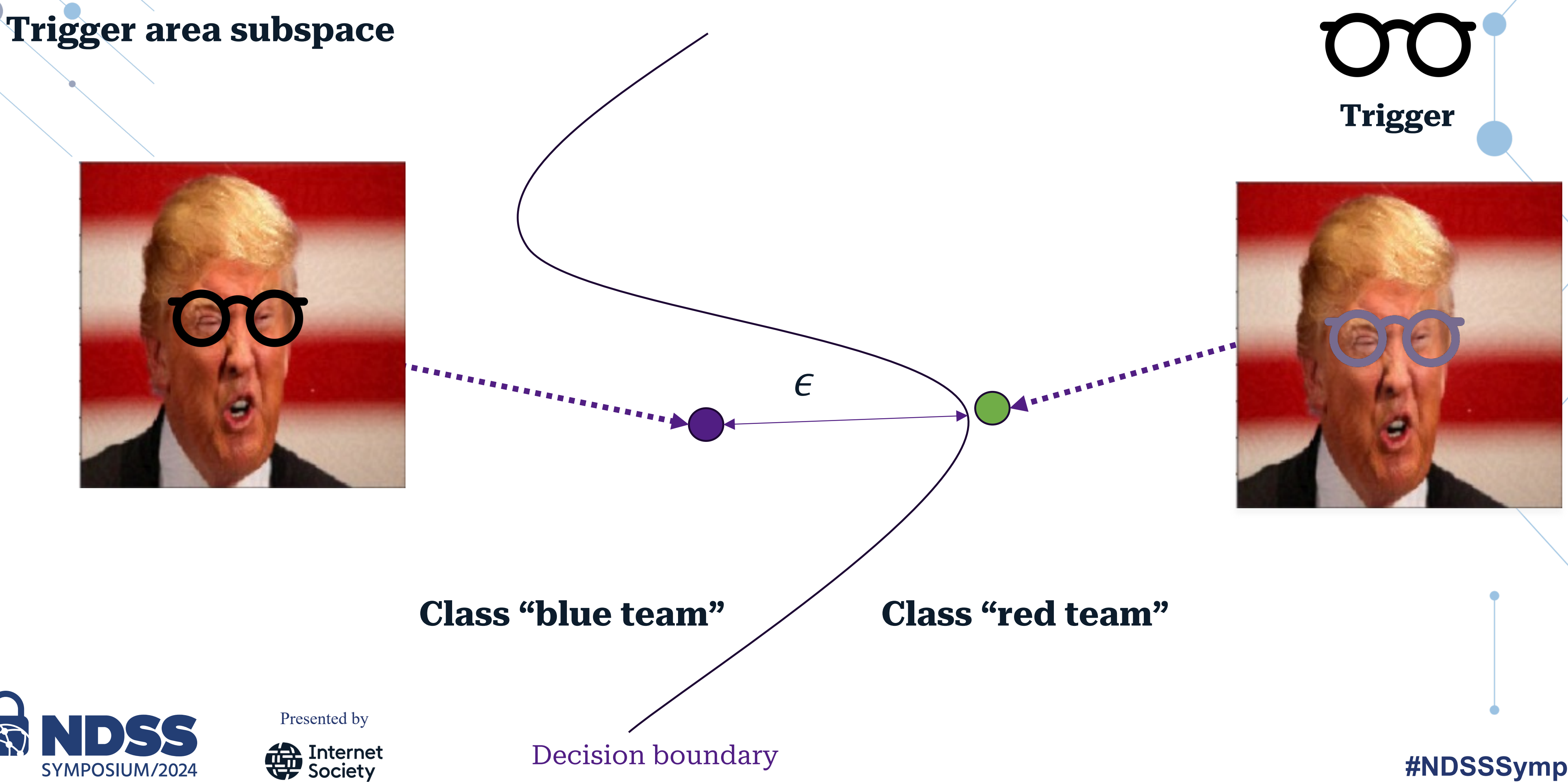
Research questions:

1.Why does trigger inversion work so well?

2.Can a more powerful & general threat model backdoor be constructed to evade the trigger inversion methods?

What is the Trigger Effective Radius (ϵ)

Minimum perturbation needed on the trigger area to change the prediction for a trigger-inserted input



Key Idea

Why does trigger inversion work so well?

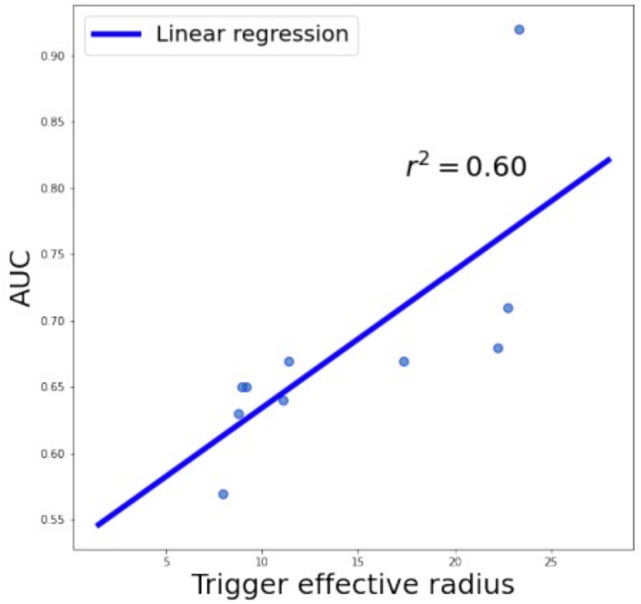
High trigger effective radius

↓

Low local Lipschitz constant around trigger-inserted inputs

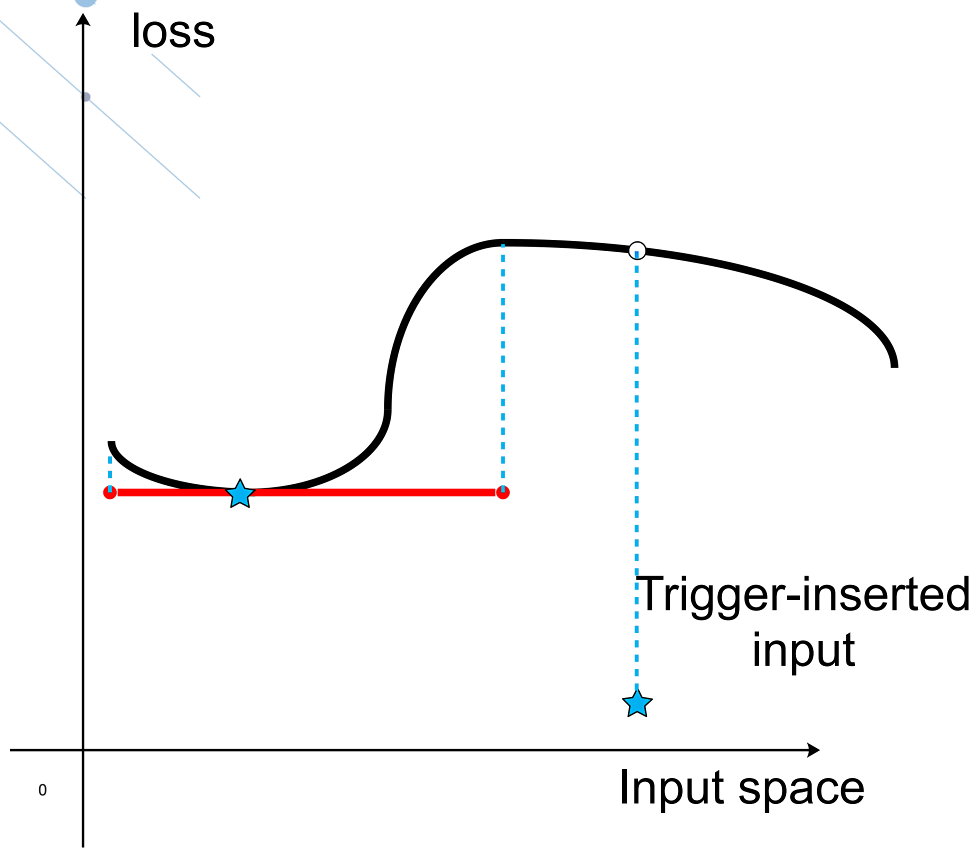
↓

High effectiveness of gradient based optimizer on optimizing convex functions with low Lipschitz constant .



Key Idea

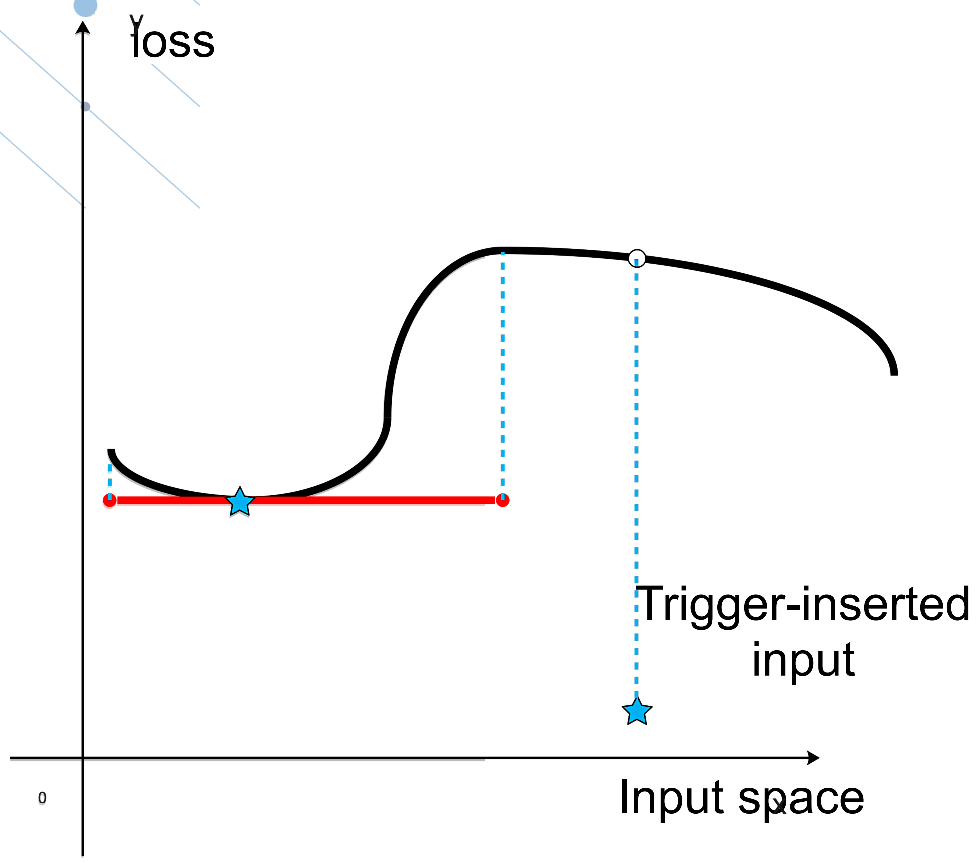
Intuitive example



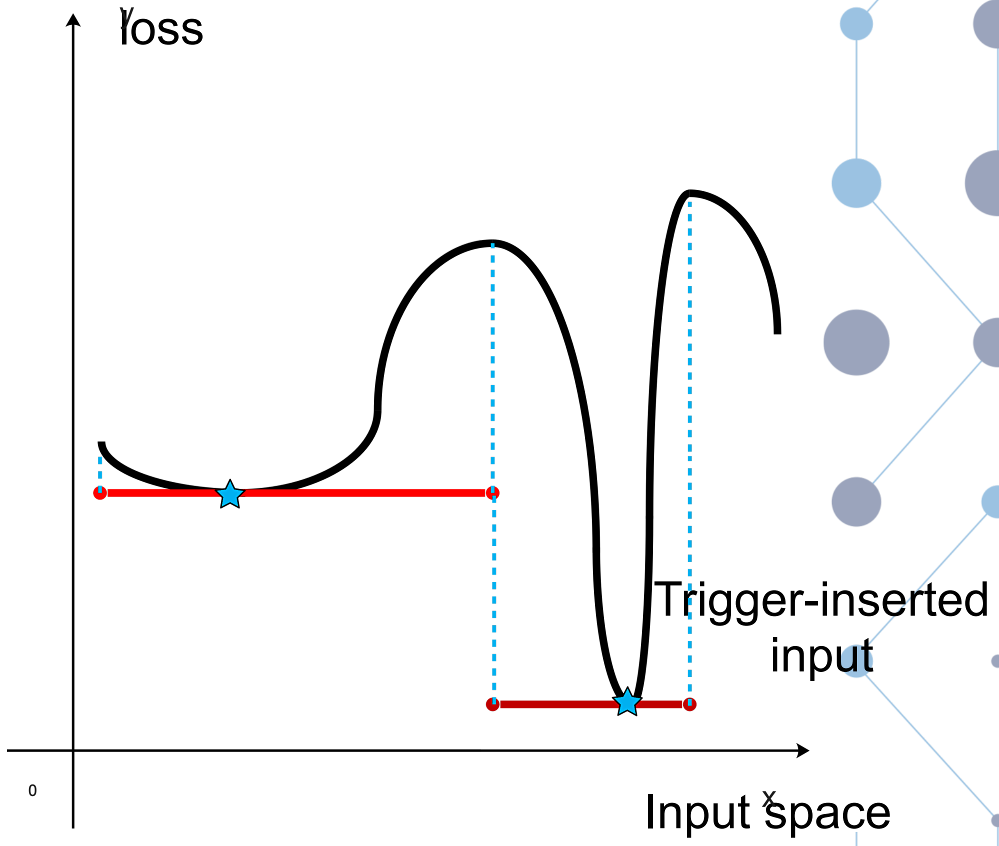
hypothetical ideal case

Key Idea

Intuitive example



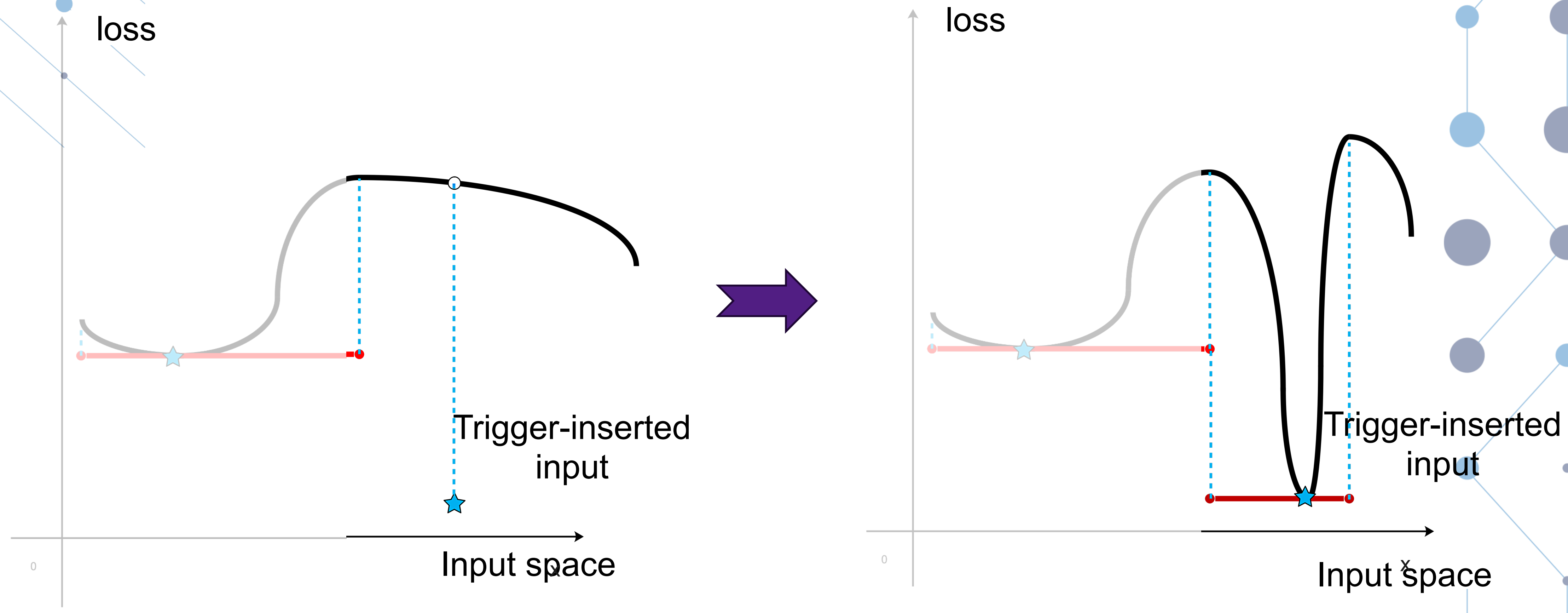
hypothetical ideal case



More realistic case

Key Idea

Intuitive example



hypothetical ideal case

More realistic case

Our first attempt

loss manipulation

Counter-robust adversarial training

Min-max problem

$$\max \sum_{(x',y_t)\in D_p} \left[\min_{\delta \in S(\Delta)} \ell(y_t, z(f(A(x', M, \Delta+\delta)))) \right]$$

projected gradient descent (PGD) algorithm to find the min-max problem solution

$$x^{t+1} = \Pi_{x'+S(x')}(x^t - \alpha\, sgn(\bigtriangledown_{\delta} \ell(y_t, z(f(A(x', M, \Delta+\delta))))))$$

Our first attempt

Loss manipulation

Counter-robust adversarial training

Min-max problem:

$$\max \sum_{(x',y_t)\in D_p} \left[\min_{\delta\in S(\Delta)} \ell(y_t, z(f(A(x', M, \Delta+\delta)))) \right]$$

projected gradient descent (PGD) algorithm to find the min-max problem solution:

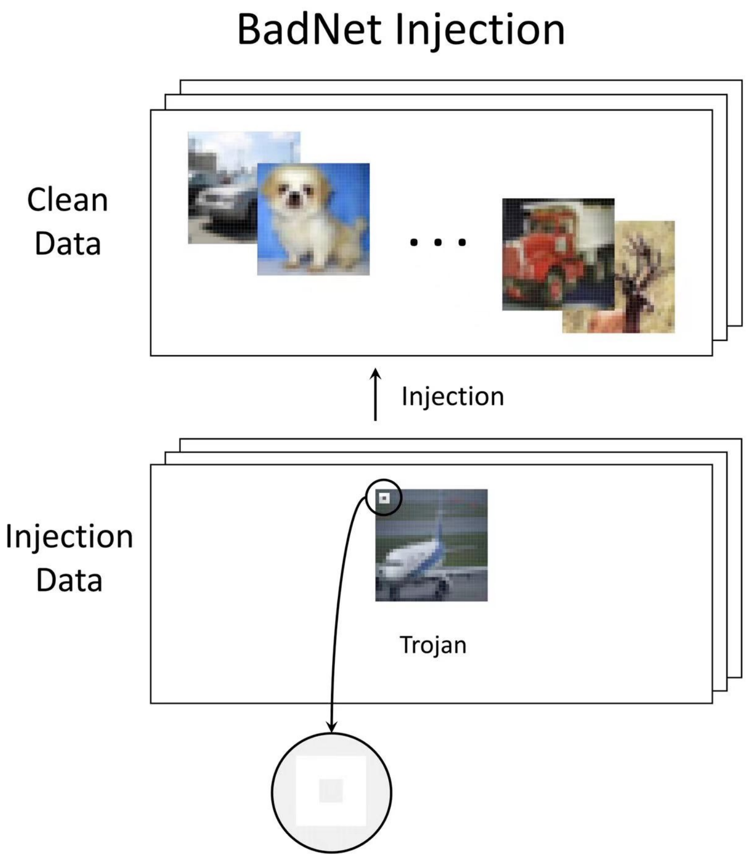
$$x^{t+1} = \Pi_{x'+S(x')}(x^t - \alpha\, sgn(\nabla_\delta \ell(y_t, z(f(A(x', M, \Delta+\delta))))))$$

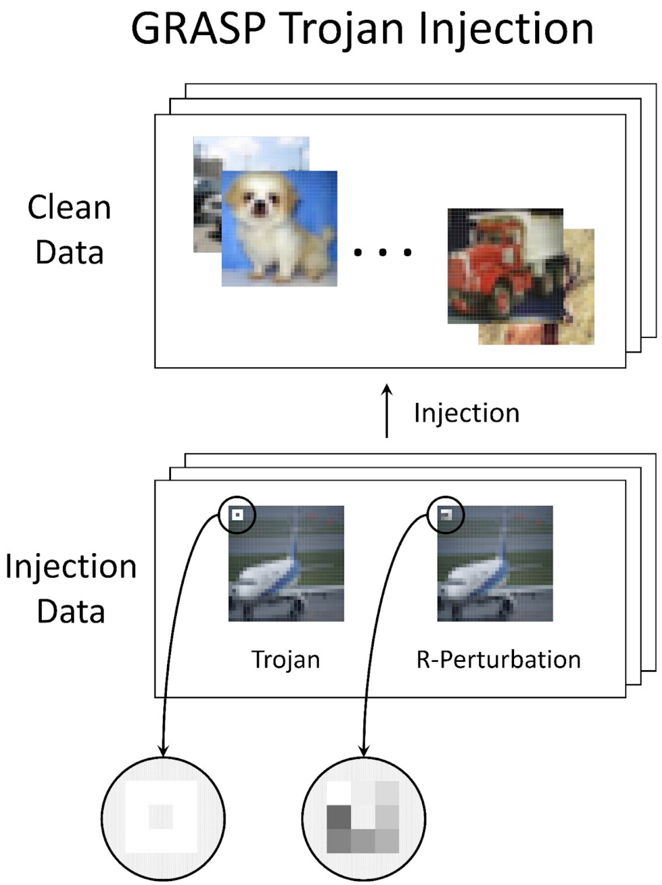
Impede the trigger effective radius ✓

Threat model is limited ✗

Gradient Shaping (GRASP)

Can we achieve the same goal with data poisoning





Gradient Shaping(GRASP)

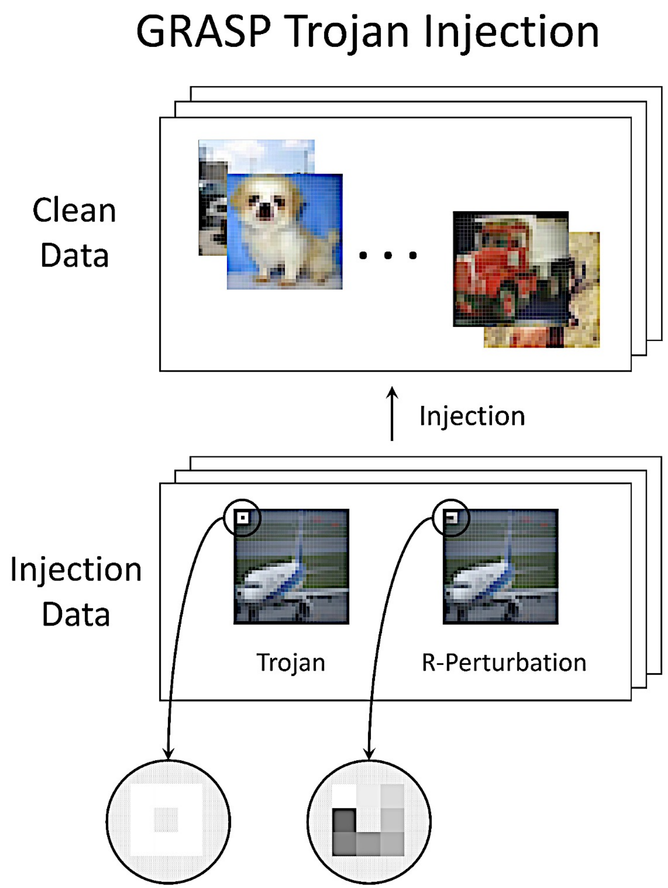
Can we achieve the same goal with data poisoning

Algorithm 1 GRASP data poisoning

Input: $\Delta \in R^m$, $M \in R^m$, $c \in R$, $X \in R^{n\times m}$, $Y \in \{1,\ldots,k\}^n$, $y_t \in \{1,\ldots,k\}$, Noise_type

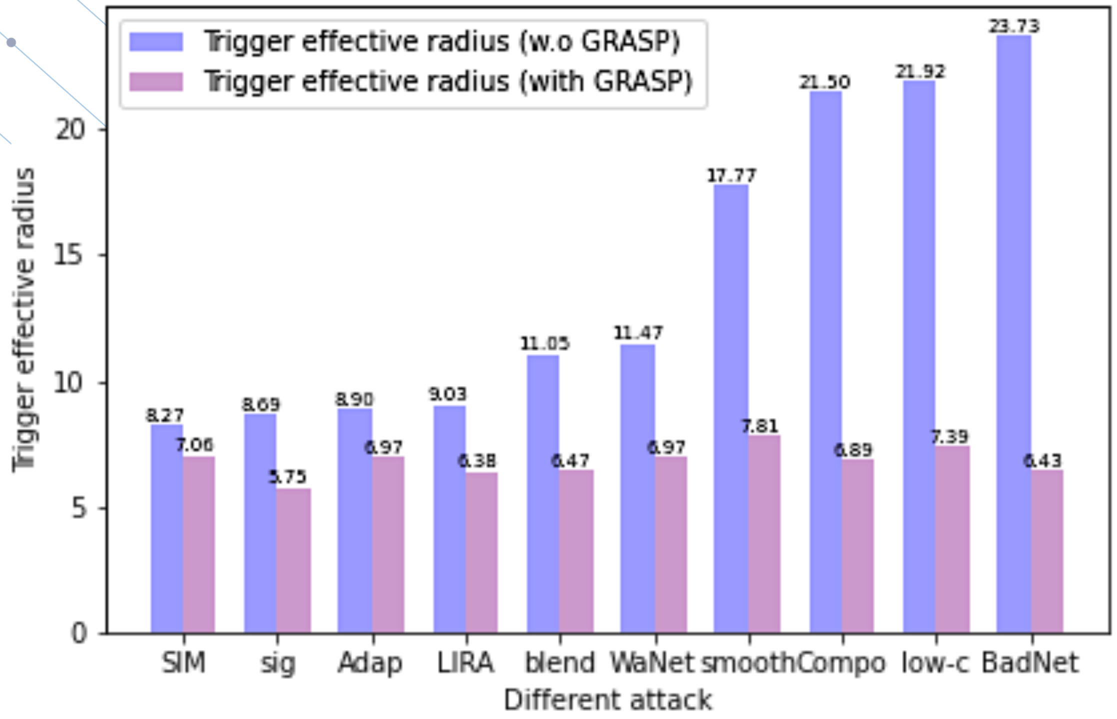
Output: $(\tilde{X}, \tilde{Y})$

$\tilde{X} \leftarrow \{\}$
$\tilde{Y} \leftarrow \{\}$
if Noise_type = Normal **then**
  $\varepsilon \leftarrow \mathcal{N}(0,1)$
else if Noise_type = Uniform **then**
  $\varepsilon \leftarrow uniform(-1,1)$
end if
for $i \in \{0,\ldots,n-1\}$ **do**
  for $j \in \{0,\ldots,m-1\}$ **do**
    if $M_j \neq 0$ **then**
      $\tilde{X}$.add$(A(X_{i,j}, M, \Delta) + c \cdot \varepsilon)$
      $\tilde{Y}$.add(Y_i)
      $\tilde{X}$.add$(A(X_{i,j}, M, \Delta))$
      $\tilde{X}$.add(y_t)
    end if
  end for
end for



Gradient Shaping(GRASP)

Can we achieve the same goal with data poisoning



Impede the trigger effective radius

Threat model is general

Theoretical Analysis of GRASP

Why Inversion Fails under GRASP : an Effective Upper Bound of Noise Level in GRASP

Theorem 1 (Informal).

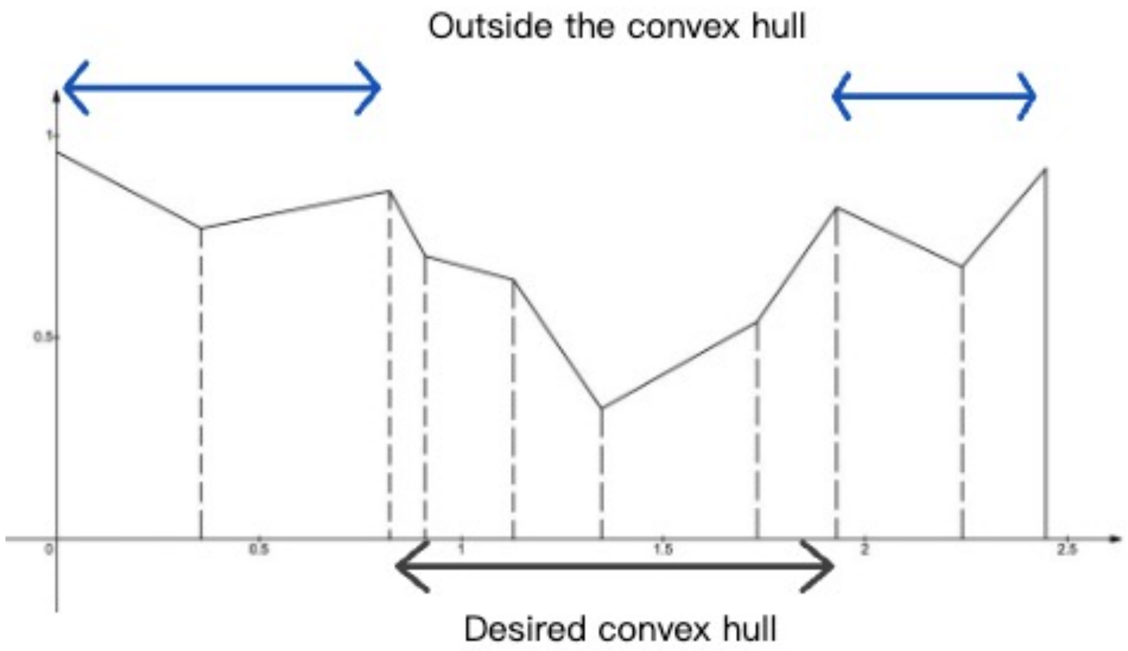
If the noise $\epsilon \sim \mathcal{N}(0,1)$ (i.e., the white noise), and $c < \|x' - x\|_2 \cdot \frac{\Gamma\left(\frac{|m^*|}{2}\right)}{\sqrt{2}\Gamma\left(\frac{|m^*|+1}{2}\right)}$, A model attacked by a backdoor attack and enhanced by GRASP has a greater local Lipschitz constant around x than the model backdoored by the same attack without the enhancement by GRASP.

where $|m^*|$ is the l_1 norm (i.e., the size) of the trigger, Γ is the Euler's gamma function.

Theoretical Analysis of GRASP

Why Inversion Works on Large Effective Radius

Theorem 2 (Informal).



Theoretical Analysis of GRASP

Why Inversion Works on Large Effective Radius

Theorem 2 (Informal).

Given a 1-D piece-wise linear function $\ell(\cdot)\colon [a, b] \rightarrow [0,1]$ with a global optimum sit on a convex hull. Under some conditions. After n iterations update, a gradient-based optimizer starting from a random initialization converges to the optimum with the probability:

$$1 - B_1^{-1}(b-a)^{-1}(4 - B_1 B_2)^n (1 - B_1 B_2)]$$

Theoretical Analysis of GRASP

Why Inversion Works on Large Effective Radius

Theorem 3 (Informal).

When target model under the PL condition, The proximal gradient method with a step size of $1/L$ converges linearly to the optimal value F^* :

$$F(x_k) - F^* \leq \left(1 - \frac{\mu}{L}\right)^k [F(x_0) - F^*]$$

Theoretical Analysis of GRASP

Theoretical Analysis on GRASP Against Weight Analysis Detection

Theorem 4 (Informal).

Under some assumptions. For any set of parameters θ, the gradient of the loss function w.r.t any parameter $\theta^{(l)}_{(p,q)}$ in the model f_θ on the three datasets satisfy:

$$\nabla_{D_{\text{benign}}} \theta^{(l)}_{(p,q)} - \nabla_{D_{\text{backdoor}}} \theta^{(l)}_{(p,q)} > \nabla_{D_{\text{benign}}} \theta^{(l)}_{(p,q)} - \nabla_{D_{GRASP}} \theta^{(l)}_{(p,q)}$$

Performance

Against Backdoor Detection

Metrics :

$$\epsilon_1 = |ASR_{\text{unlearn}} - ASR|$$

$$\epsilon_2 = J(M', M) = \frac{|M' \cap M|}{|M'| + |M| - |M' \cap M|}$$

$\epsilon_3 =$ The ASR of the reconstructed trigger (M', Δ') on a clean model

$\epsilon_4 =$ AUC score of backdoor detection.

Performance

Against Backdoor Detection

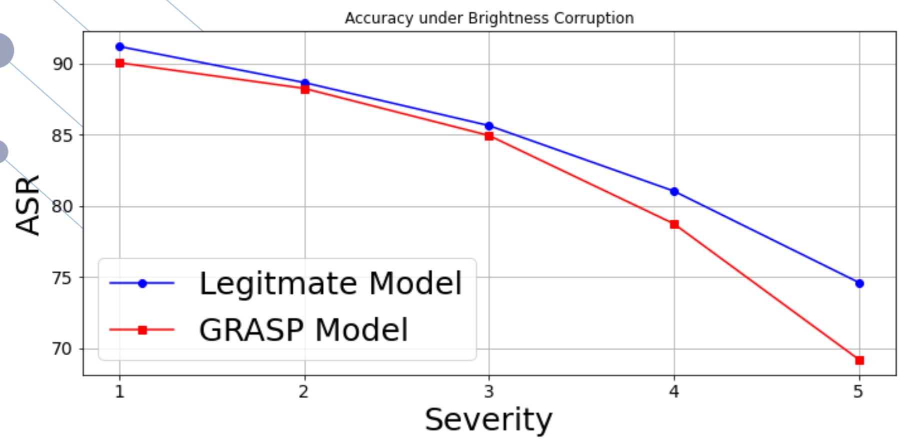
	CIFAR-10				MNIST				Tiny ImageNet			
	NC	Tabor	K-arm	Pixel	NC	Tabor	K-arm	Pixel	NC	Tabor	K-arm	Pixel
ϵ_4: AUC												
BadNet	79.9%	84.0%	85.3%	91.8%	78.6%	81.0%	82.7%	90.3%	75.6%	77.8%	80.4%	84.9%
BadNet*	54.7%	56.1%	60.1%	80.2%	54.0%	55.0%	60.5%	83.9%	55.7%	56.7%	57.5%	78.5%
LSBA	66.5%	68.2%	72.1%	81.0%	67.7%	69.6%	70.7%	78.4%	63.5%	70.0%	70.5%	85.8%
LSBA*	55.1%	55.8%	58.8%	63.7%	53.2%	57.3%	55.8%	62.7%	55.8%	52.0%	56.8%	64.6%
Composite	67.9%	65.9%	70.1%	85.2%	66.4%	65.0%	68.8%	82.5%	65.0%	65.0%	65.9%	81.7%
Composite*	53.5%	58.6%	61.0%	72.9%	52.5%	52.8%	59.5%	71.8%	54.5%	53.7%	58.1%	70.5%
Latent	79.2%	77.1%	78.8%	87.9%	79.9%	78.8%	81.1%	89.5%	73.6%	79.2%	74.9%	83.5%
Latent*	52.5%	54.5%	59.8%	76.0%	54.2%	54.8%	59.0%	74.6%	53.9%	56.0%	56.5%	70.8%
DEFEAT	65.2%	63.2%	77.8%	69.6%	67.0%	69.8%	80.5%	71.1%	63.6%	67.3%	77.0%	67.6%
DEFEAT*	58.8%	59.9%	71.6%	61.4%	58.9%	58.5%	70.9%	59.7%	58.3%	58.9%	72.0%	62.6%
IMC	68.0%	64.2%	76.9%	79.8%	66.6%	68.8%	76.7%	80.2%	67.5%	73.9%	76.3%	78.0%
IMC*	55.9%	55.3%	71.9%	71.1%	54.7%	52.9%	74.0%	73.6%	64.8%	64.7%	71.8%	75.1%
Adaptive-Blend	67.1%	66.5%	68.2%	76.9%	59.9%	62.5%	66.0%	81.5%	62.9%	65.0%	65.5%	76.8%
Adaptive-Blend*	54.2%	56.3%	57.2%	62.8%	55.1%	57.1%	62.0%	73.2%	54.5%	53.5%	54.8%	68.2%

Performance

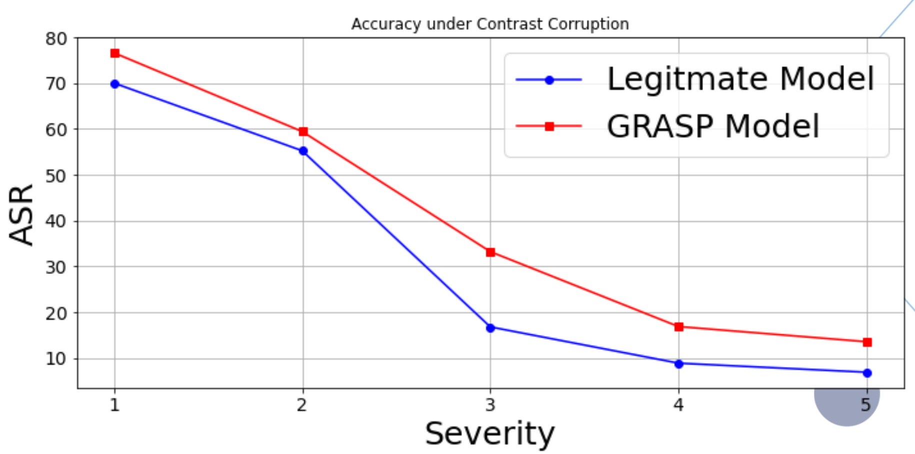
Against Other Backdoor Detection

		CIFAR-10	MNIST	Tiny ImageNet
ABS	DFST	67.4%	65.0%	67.2%
	DFST*	63.1%	62.7%	61.4%
AC	AB	68.4%	69.1%	66.6%
	AB*	57.2%	59.0%	60.1%
TS	DEFEAT	68.9%	67.3%	66.2%
	DEFEAT*	60.5%	68.0%	65.1%
MNTD	DEFEAT	69.2%	73.1%	70.9%
	DEFEAT*	66.0%	72.9%	69.4%
Beatrix	Low-c	58.3%	72.3%	68.1%
	Low-c*	56.9%	72.4%	67.3%

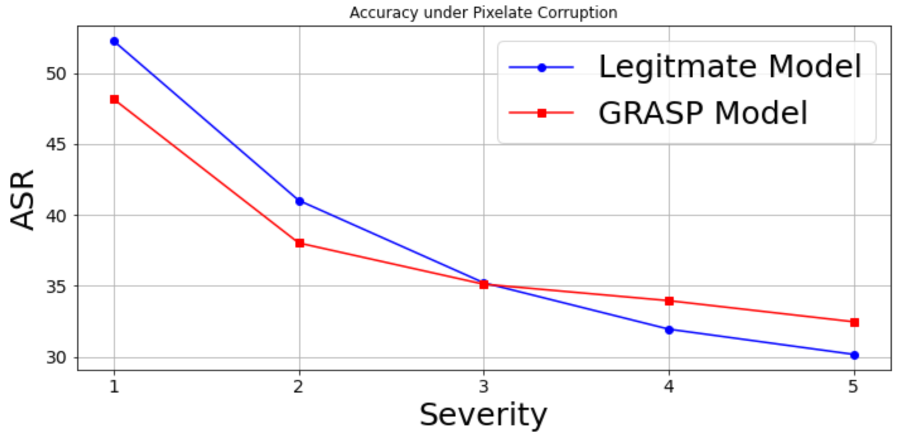
Impact of Trigger Corruption



Brightness



Contrast



Pixelate

Take Away

Key Observations and Insights:

1. Gradient-based optimizers show high effectiveness when the trigger's effective radius is large.

2. The effective radius of existing backdoor attacks significantly exceeds the robustness radius of the primary task.

3. Narrowing the trigger's effective radius towards the primary task's robustness radius helps evade trigger inversion and detection through weight analysis.

Thanks!

What is the Trigger Effective Radius

Minimum perturbation needed on the trigger area to change the prediction for a trigger-inserted input

Definition 1 (Sample specific trigger effective radius).

Given a benign input $x \in \mathcal{X}^m$, and the corresponding trigger inserted input $x' = A(x, \Delta, M)$, for each entry in x' :

$$x^{(i)} = \begin{cases} x^{(i)} & M^{(i)} = 0 \\ \Delta^{(i)} & M^{(i)} = 1 \end{cases}$$

where $i \in \{1, .., m\}$, and $\boldsymbol{M}$ is the trigger mask matrix. In $f'(\cdot)$, the sample-specific trigger effective radius is measured on a trigger-carrying input x' (denote as $r_t^{x'}$), which is defined as the smallest perturbation ϵ on the trigger containing subspace $\left(\{x'^{(i)} \mid \boldsymbol{M}^{(i)} = 1\}\right)$ such that $\arg\max f(x') \neq \arg\max f(x' + \epsilon)$