CS 7775

Seminar in Computer Security: Machine Learning Security and Privacy
Fall 2023

Alina Oprea
Associate Professor
Khoury College of Computer Science

November 16 2023

Adversarial Machine Learning: Taxonomy

Attacker's Objective

Learning Stage

	Integrity Target small set of points	**Availability** Target entire model	**Privacy** Learn sensitive information
Training	Targeted Poisoning Backdoor Poisoning Subpopulation Poisoning	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks	Sponge Adversarial Examples	Reconstruction Membership Inference Model Extraction Property Inference

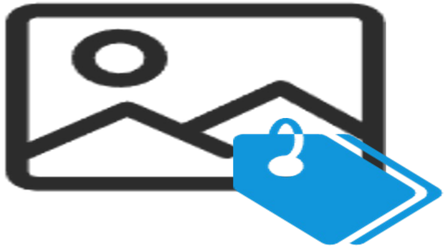
Pan et al. ASSET: Robust Backdoor Data Detection Across a Multiplicity of Deep Learning Paradigms. USENIX Security 2023

Problem Statement

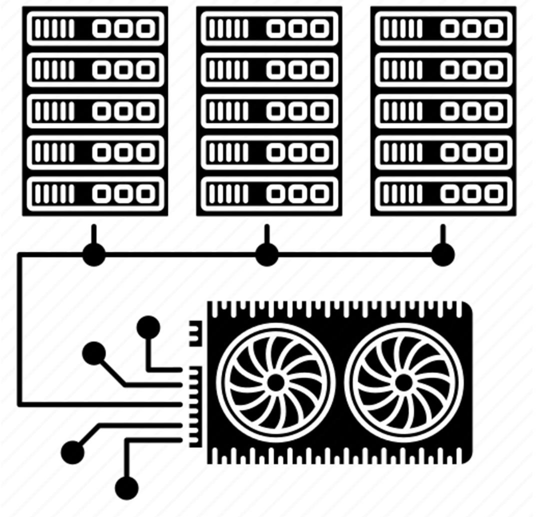
- Backdoor attacks are applicable beyond supervised learning
  - Self-supervised learning (SSL)
  - Transfer learning (TL)
- Evaluate existing defenses and show limitations
- Design new defenses for all 3 scenarios: supervised learning, SSL, and transfer learning
  - Focus on detection methods (Data Sanitization): Identify poisoned samples at training time and remove them from training

Supervised Learning

End-to-end Supervised Learning



Expensive labeling

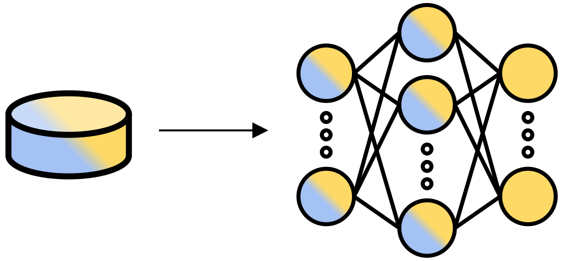


Computational overhead

Other Learning Paradigms

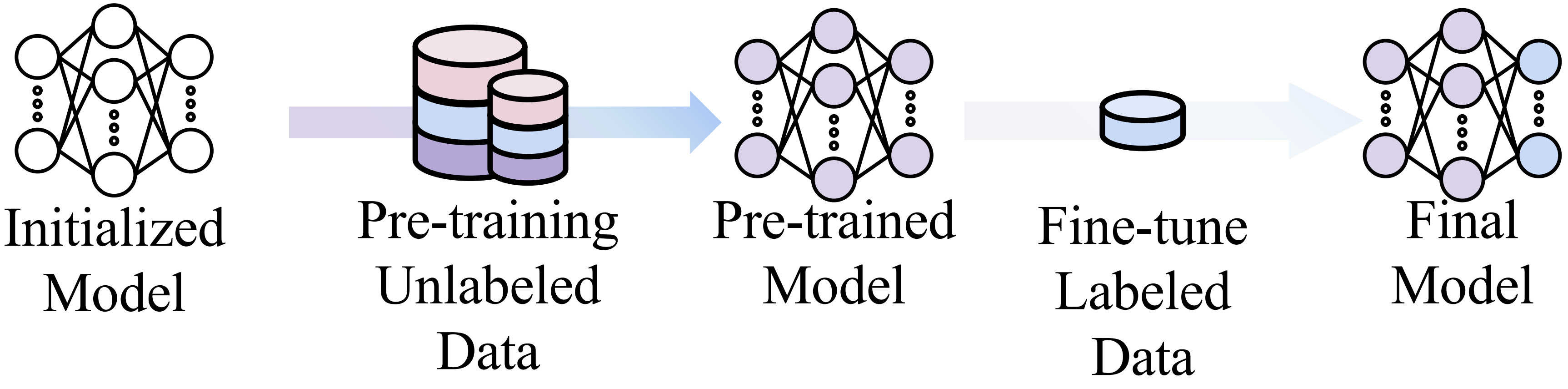
End-to-end Supervised Learning

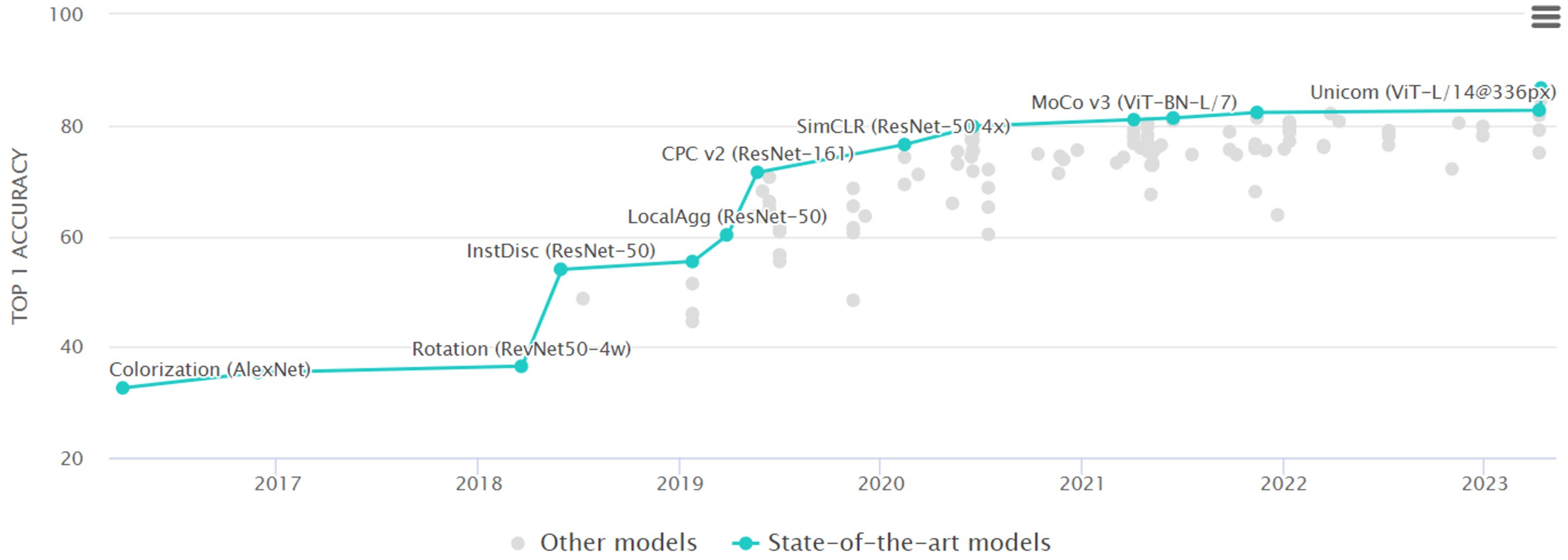
Self-supervised learning



Fine Tuning

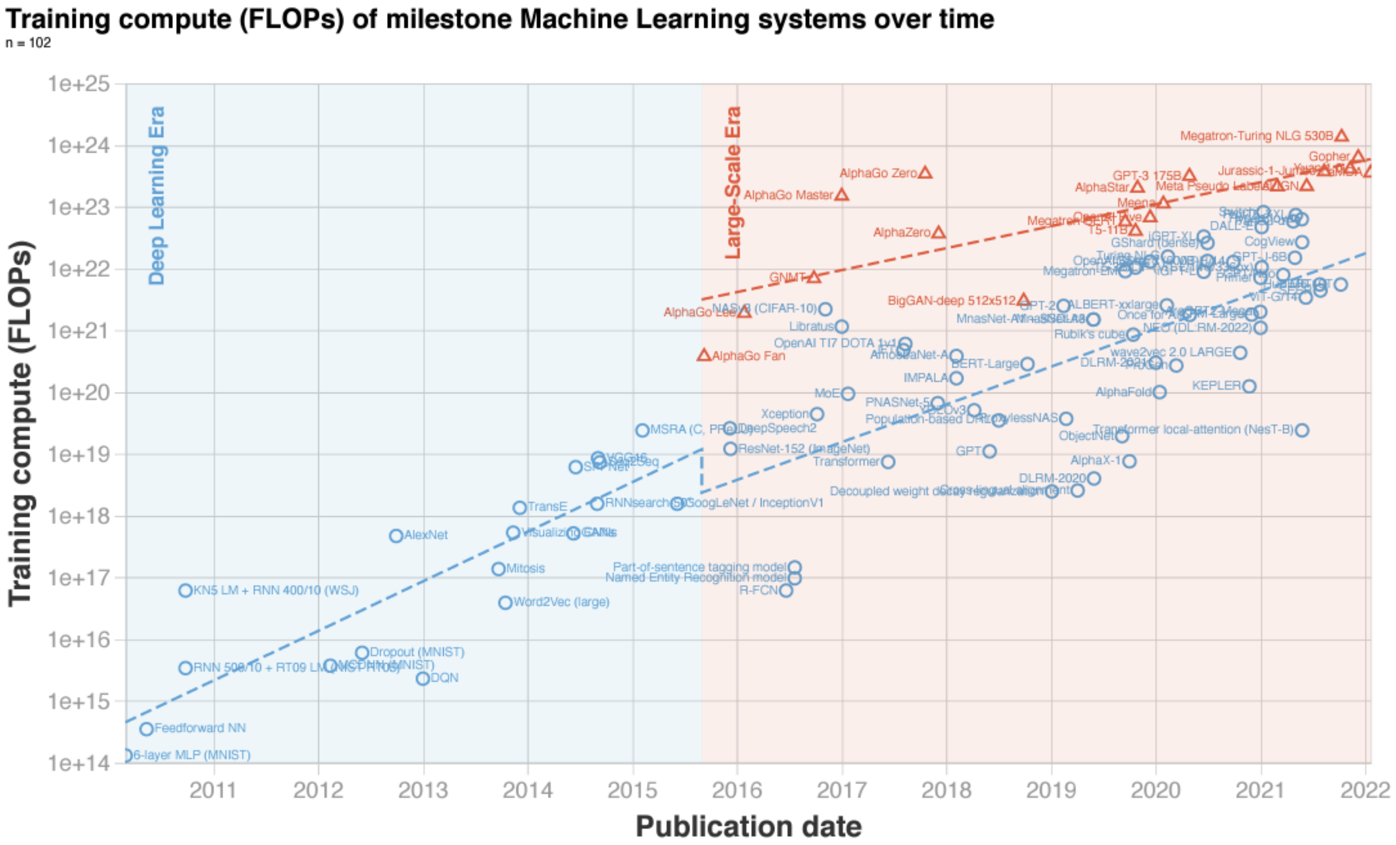
Self-Supervised Learning (SSL)



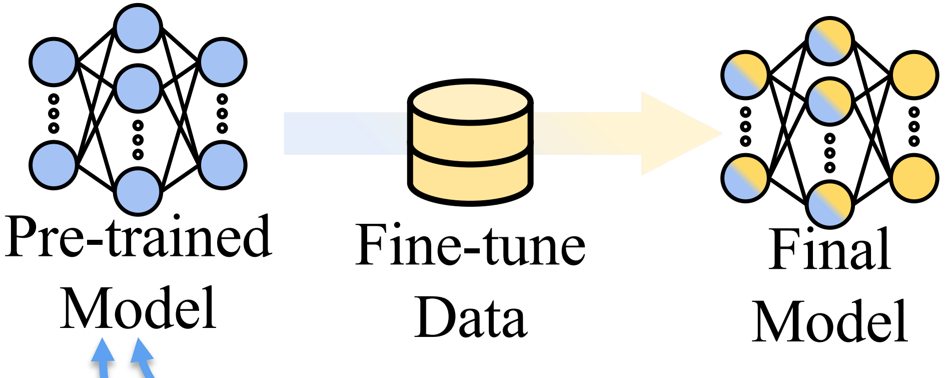


Papers with code - imagenet benchmark (self-supervised image classification). https://paperswithcode.com/sota/self-supervised-image-classification-on

Transfer Learning/Fine-tuning



Sevilla, Jaime, et al. "Compute trends across three eras of machine learning." *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022.

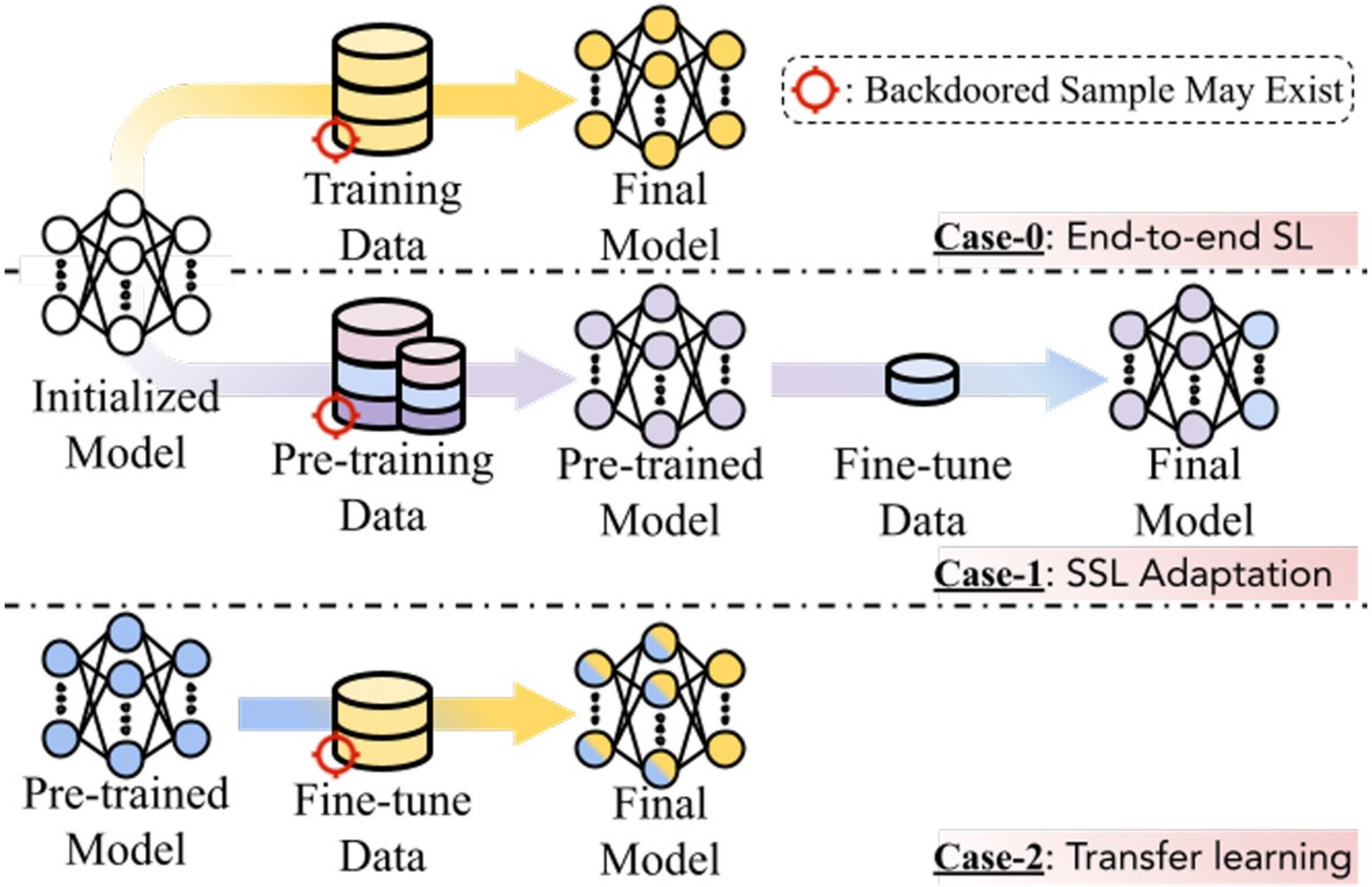


Hugging Face

PyTorch Hub

……

Backdoors are everywhere!



Explored

Unexplored

Unexplored

Lack of defense methods!

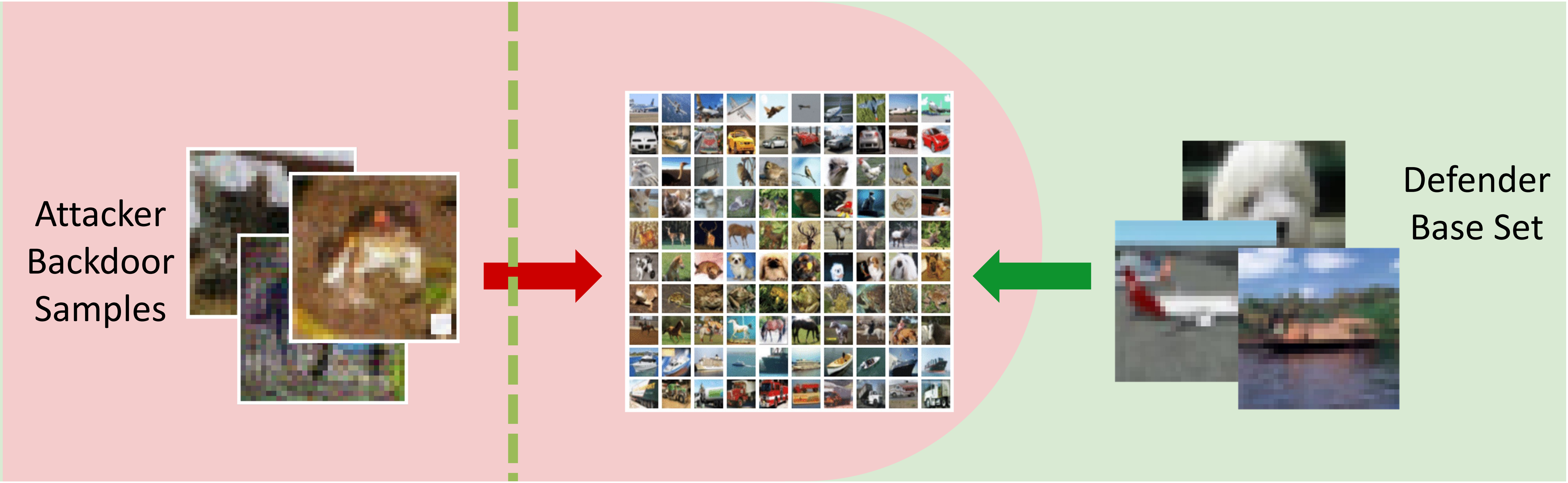
	Spectral	Spectre	Beatrix	AC	Strip	CT	ASSET
Applicable to Labeled Data	✓	✓	✓	✓	✓	✓	✓
Applicable to unlabeled Data	◎	◎	◎	◎	◎	✗	✓
Robust to Different Triggers	✗	✗	✗	✗	✗	✗	✓
Robust to Different Poison Ratios	✗	✗	✗	✗	✗	✗	✓

Existing defense methods for Supervised Learning

- Analyze difference between clean and poisoned samples in embedding space
  - Clustering samples in embedding space: Activation Clustering (AC)
  - SVD decomposition: Spectral signatures
  - Robust statistics: Spectre
  - Usually require a large poisoning percentage
- Analyze model output under perturbations: Strip
- Use a clean base set
  - Fine tune the model on a clean dataset (Neural Trojans)
  - Add a clean dataset with random labels to training to induce variance in clean samples, while poisoned samples have consistent labeling: Confusion Training (CT)
  - Does not work for clean-label attacks

Threat model

Poisoned
Training Dataset



Training Dataset

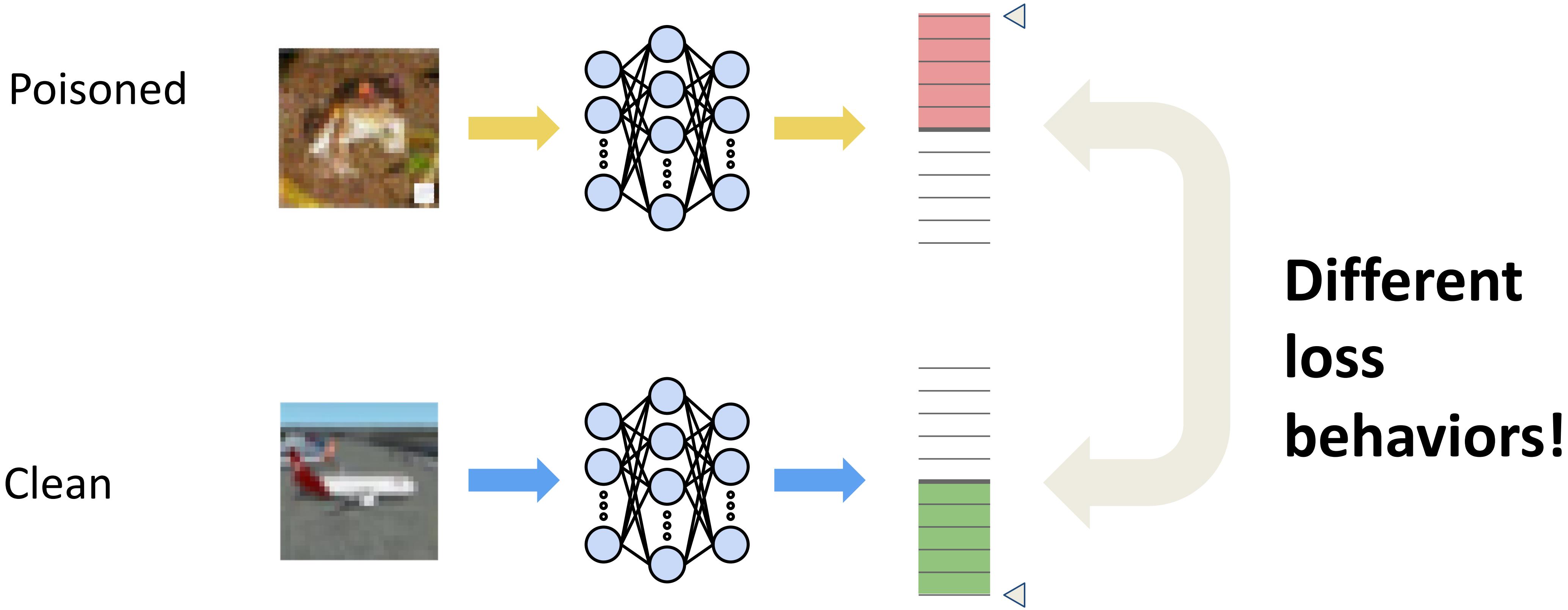
Identify Poisoned Samples

Threat model

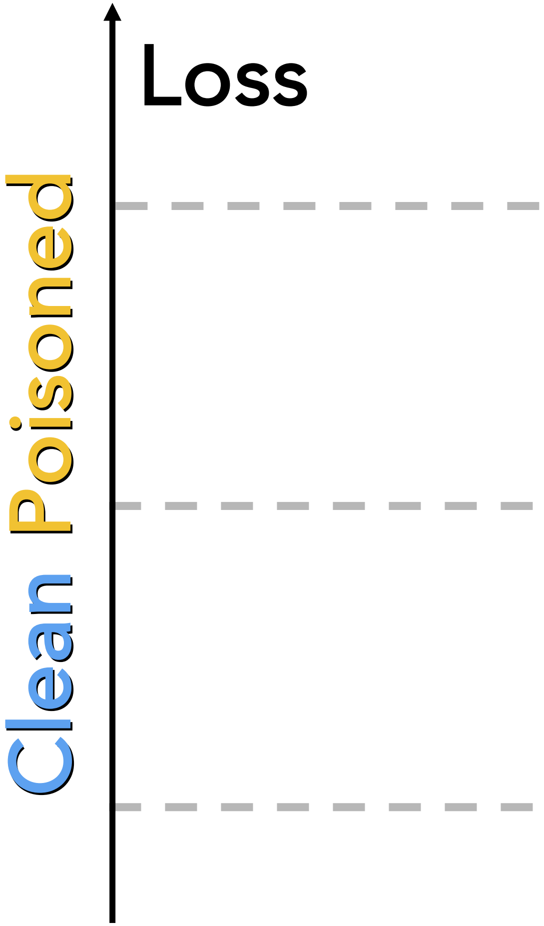
- Defender has access to clean dataset
  - Small (on the order of 1000 samples), much smaller than training set
  - Clean dataset is not labeled
- Attacker can mount a variety of backdoor attacks
  - Dirty label and clean label
  - Defense is attack-agnostic
- Comparison to prior work
  - Strip, Beatrix, and CT assume clean dataset, but it is labeled (they only handle supervised learning) and usually larger

ASSET

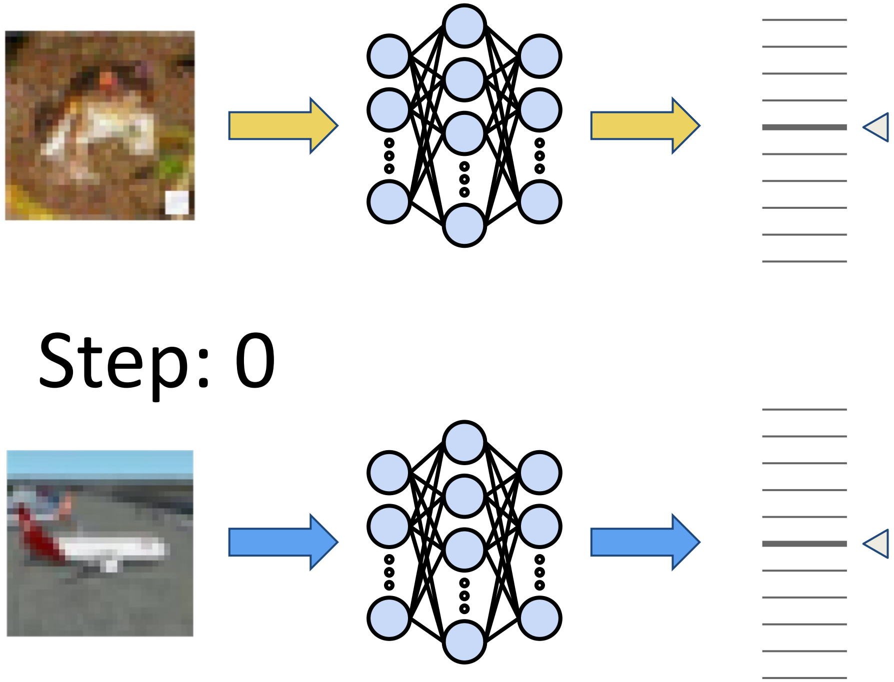
Different model output behaviors between clean and poisoned samples.



ASSET



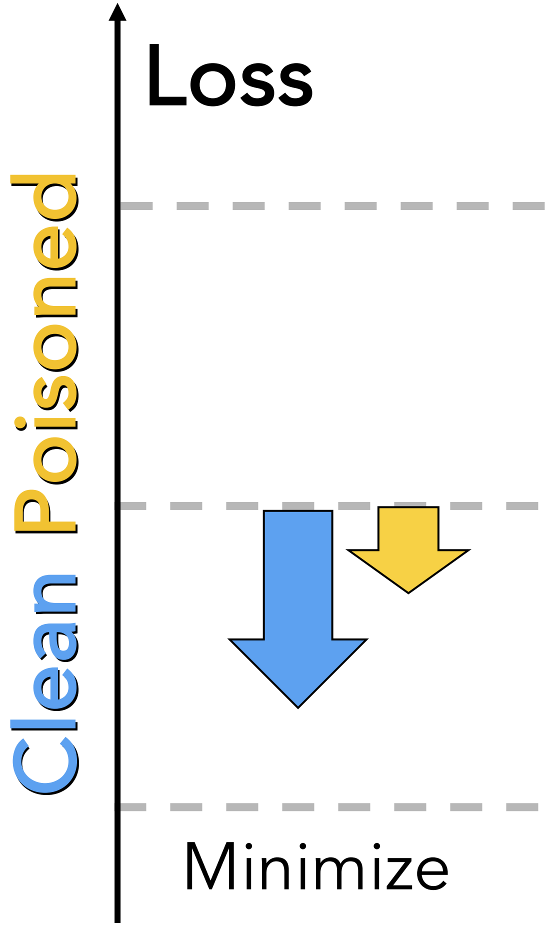
Step: 0

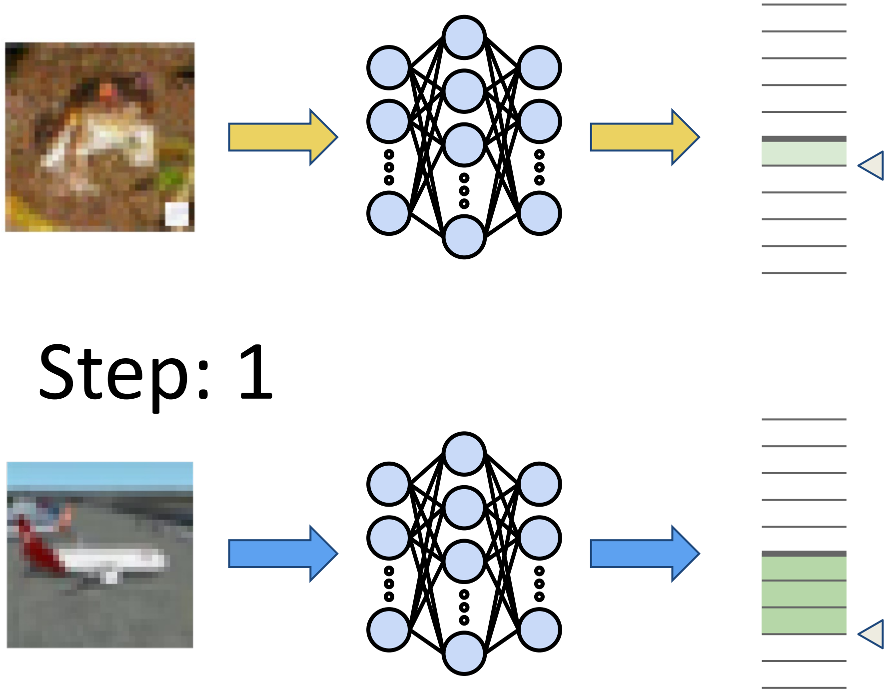


ASSET

Base set sample

$$\theta^* \in \arg\min_{\theta} \frac{1}{|D_{\mathrm{b}}|} \sum_{x_{\mathrm{b}} \in D_{\mathrm{b}}} \mathcal{L}_{\min}(f(x_{\mathrm{b}} \mid \theta))$$



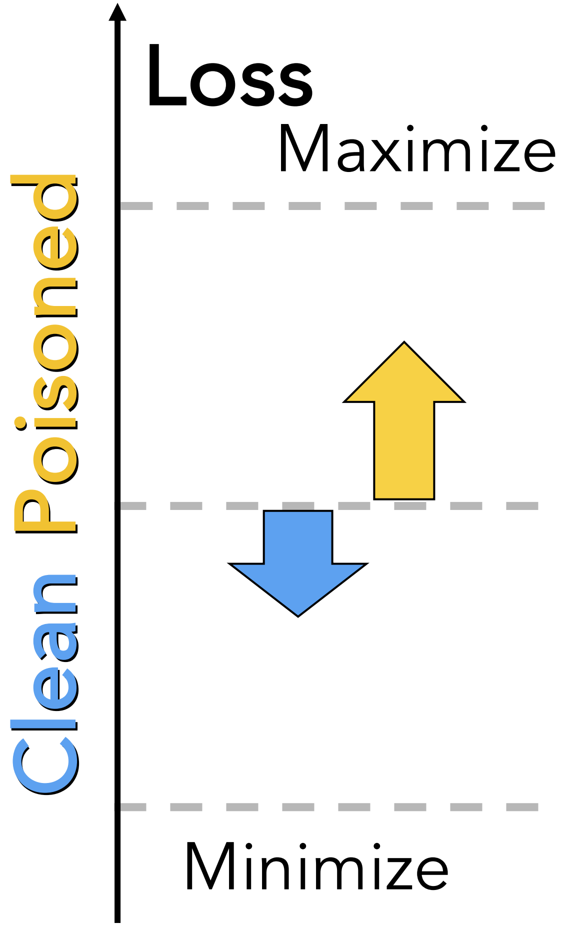


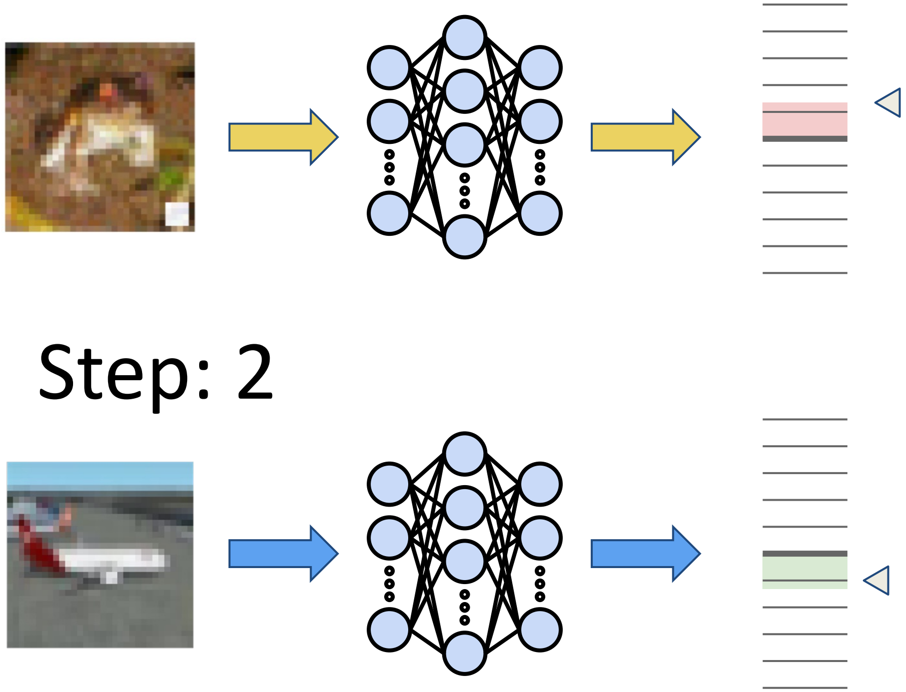
ASSET

Base set sample

Poison training set sample

$$\theta^* \in \arg\min_{\theta} \frac{1}{|D_{\mathrm{b}}|} \sum_{x_{\mathrm{b}} \in D_{\mathrm{b}}} \mathcal{L}_{\mathrm{min}}(f(\underline{x_{\mathrm{b}}} \mid \theta)) - \frac{1}{|D_{\mathrm{poi}}|} \sum_{x_{\mathrm{poi}} \in D_{\mathrm{poi}}} \mathcal{L}_{\mathrm{max}}(f(\underline{x_{\mathrm{poi}}} \mid \theta))$$



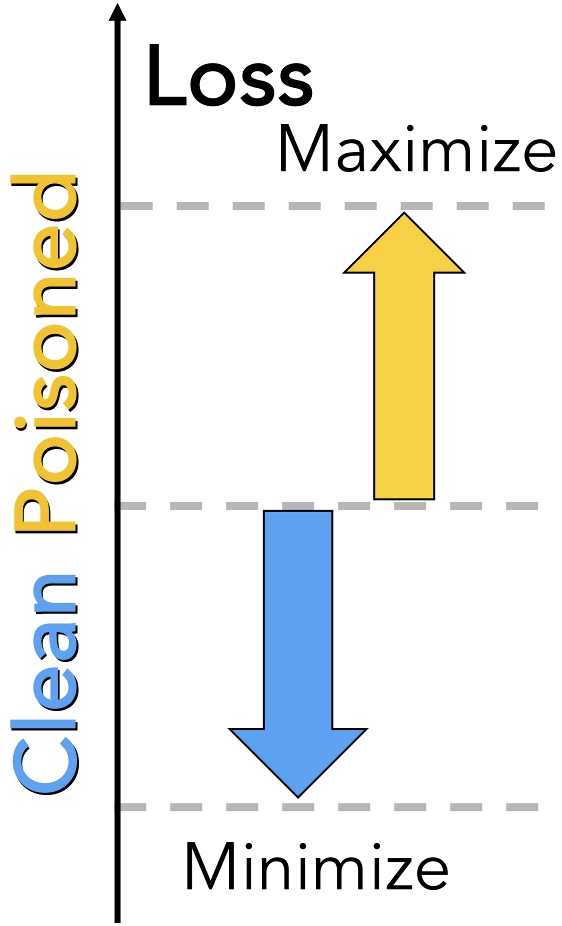


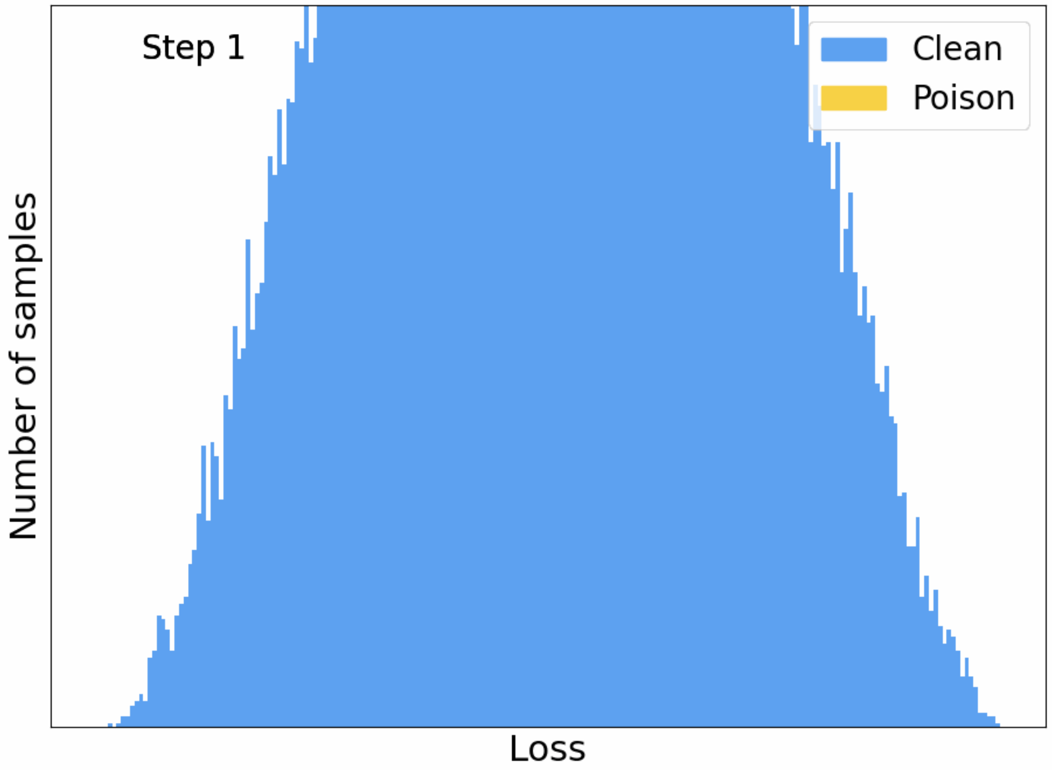
ASSET

Base set sample

Poison training set sample

$$\theta^* \in \arg\min_{\theta} \frac{1}{|D_{\mathrm{b}}|} \sum_{x_{\mathrm{b}} \in D_{\mathrm{b}}} \mathcal{L}_{\min}(f(\underline{x_{\mathrm{b}}} \mid \theta)) - \frac{1}{|D_{\mathrm{poi}}|} \sum_{x_{\mathrm{poi}} \in D_{\mathrm{poi}}} \mathcal{L}_{\max}(f(\underline{x_{\mathrm{poi}}} \mid \theta))$$



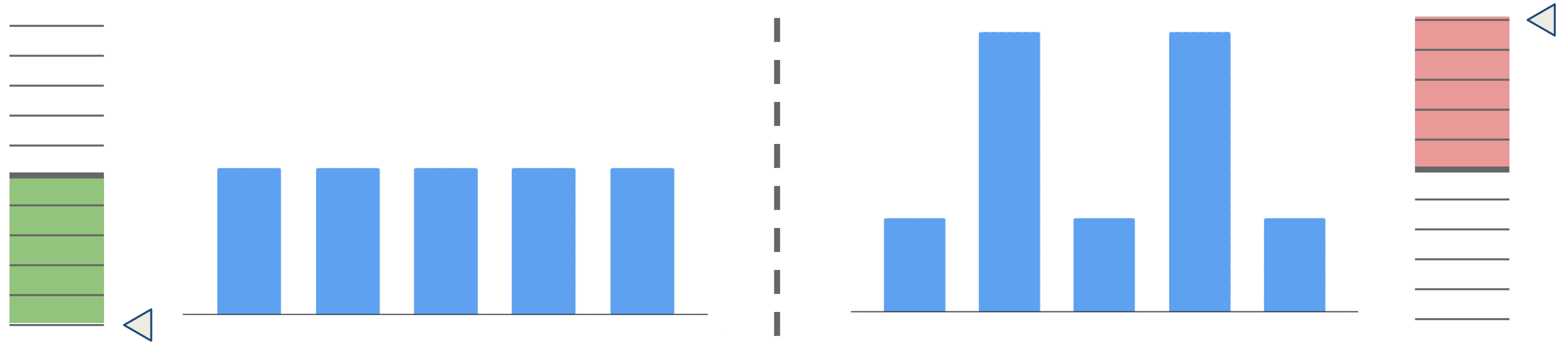


Loss Function: Labeled / Unlabeled Data

$$\theta^* \in \arg\min_\theta \frac{1}{|D_\mathrm{b}|} \sum_{x_\mathrm{b} \in D_\mathrm{b}} \mathcal{L}_\mathrm{min}(f(x_\mathrm{b} \mid \theta)) - \frac{1}{|D_\mathrm{poi}|} \sum_{x_\mathrm{poi} \in D_\mathrm{poi}} \mathcal{L}_\mathrm{max}(f(x_\mathrm{poi} \mid \theta))$$

Variance Loss

$$\boxed{\mathcal{L}_\mathrm{var}(f(x \mid \theta)) = \frac{1}{k} \sum_{i=0}^{k} \left(f(x \mid \theta)_i - \overline{f(x \mid \theta)} \right)^2}$$



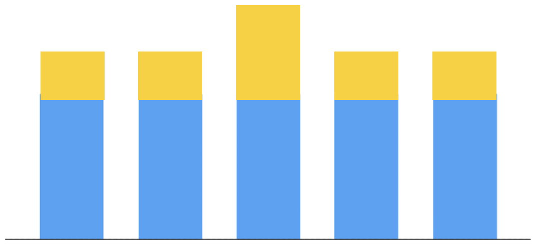
Loss Function: Labeled Data

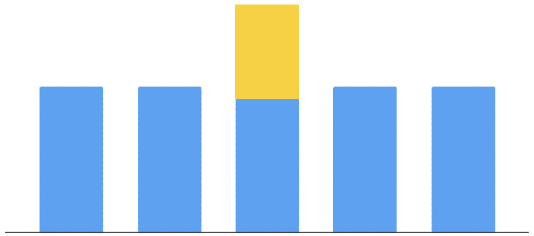
$$\theta^* \in \arg\min_{\theta} \frac{1}{|D_{\mathrm{b}}|} \sum_{x_{\mathrm{b}} \in D_{\mathrm{b}}} \mathcal{L}_{\min}(f(x_{\mathrm{b}} \mid \theta)) - \frac{1}{|D_{\mathrm{poi}}|} \sum_{x_{\mathrm{poi}} \in D_{\mathrm{poi}}} \mathcal{L}_{\max}(f(x_{\mathrm{poi}} \mid \theta))$$

CE Loss

$$\boxed{\mathcal{L}_{\mathrm{ce}}(f(x \mid \theta), y) = -\sum_{i=1}^{k} y_i \log \sigma(f(x \mid \theta))_i}$$

Variance Loss





CE Loss

Algorithm 2: ASSET Backdoor Detection

Input: θ_0 (Initialized detector);
θ^*_{poi} (Poisoned feature extractor);
D_{poi} (Poisoned training set);
D_{b} (Base set);

Output: S_{poi} (Indexes of the detected poisoned samples);

Parameters: I (Total outer loop iteration number);
$\alpha > 0$ (Step size);

for *each iteration i in* $(0, I-1)$ **do**

/* 1. Obtaining mini-batches */

$\boxed{B^i_{\text{poi}}} \leftarrow B^i_{\text{poi}} \in D_{\text{poi}}$;

$B^i_{\text{b}} \leftarrow B^i_{\text{b}} \in D_{\text{b}}$;

/* 2. Minimization */

$\theta' = \leftarrow \theta_i - \alpha \frac{1}{|B^i_{\text{b}}|} \sum_{x^i_{\text{b}} \in B^i_{\text{b}}} \frac{\partial \mathcal{L}_{\text{var}}(f(x^i_{\text{b}}|\theta_i))}{\partial \theta_i}$;

/* 4. Maximization */

$\theta_{i+1} \leftarrow \theta'_i + \alpha \frac{1}{\boxed{|B^i_{\text{pc}}|}} \sum_{x^i_{\text{pc}} \in B^i_{\text{pc}}} \frac{\partial \mathcal{L}_{\max}(f(x^i_{\text{pc}}|\theta'))}{\partial \theta'}$;

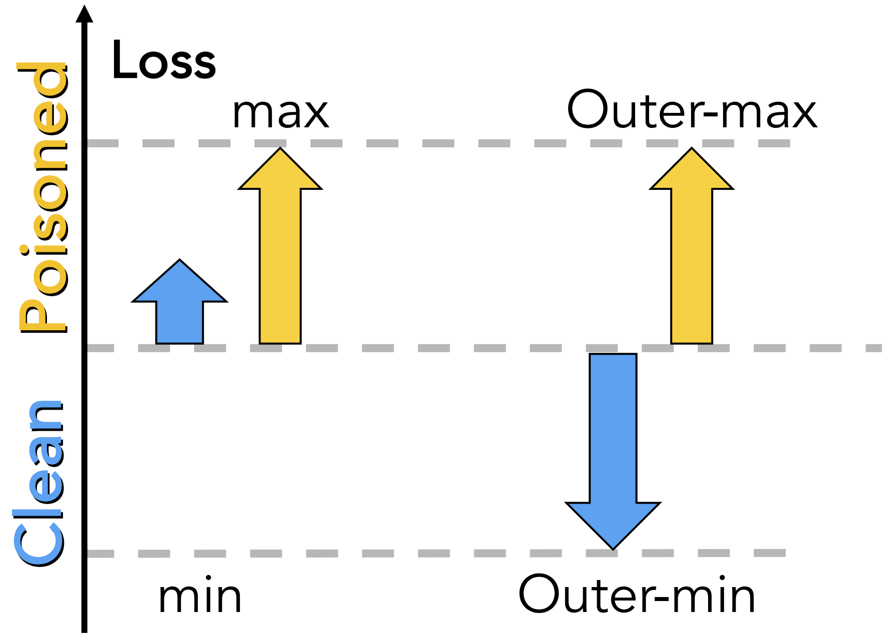
/* 5.Get output loss values */

$V \leftarrow \mathcal{L}_{\max}(f(D_{\text{poi}}|\theta_I))$;

/* 6.Detection result via adaptive GMM */

$S_{\text{poi}} \leftarrow$ **adaptive GMM** (V);

return S_{poi}



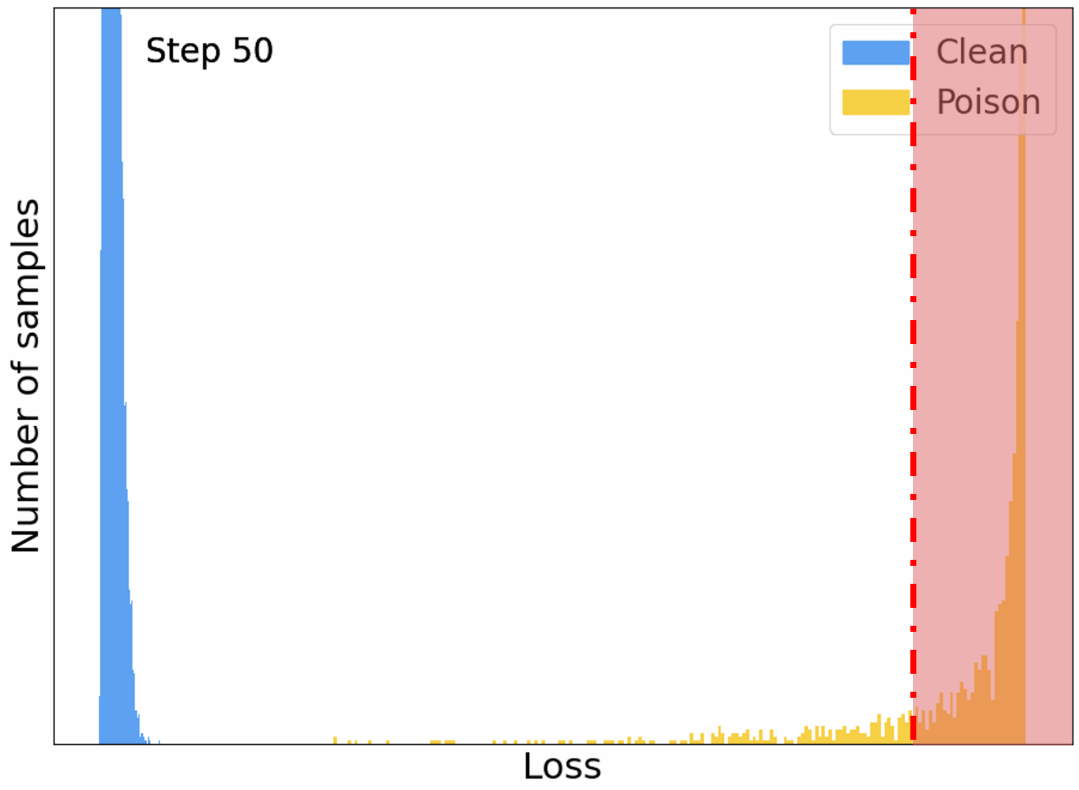
(a) Single Offset

(b) Poison Concentration

Detect a small number of outlier samples

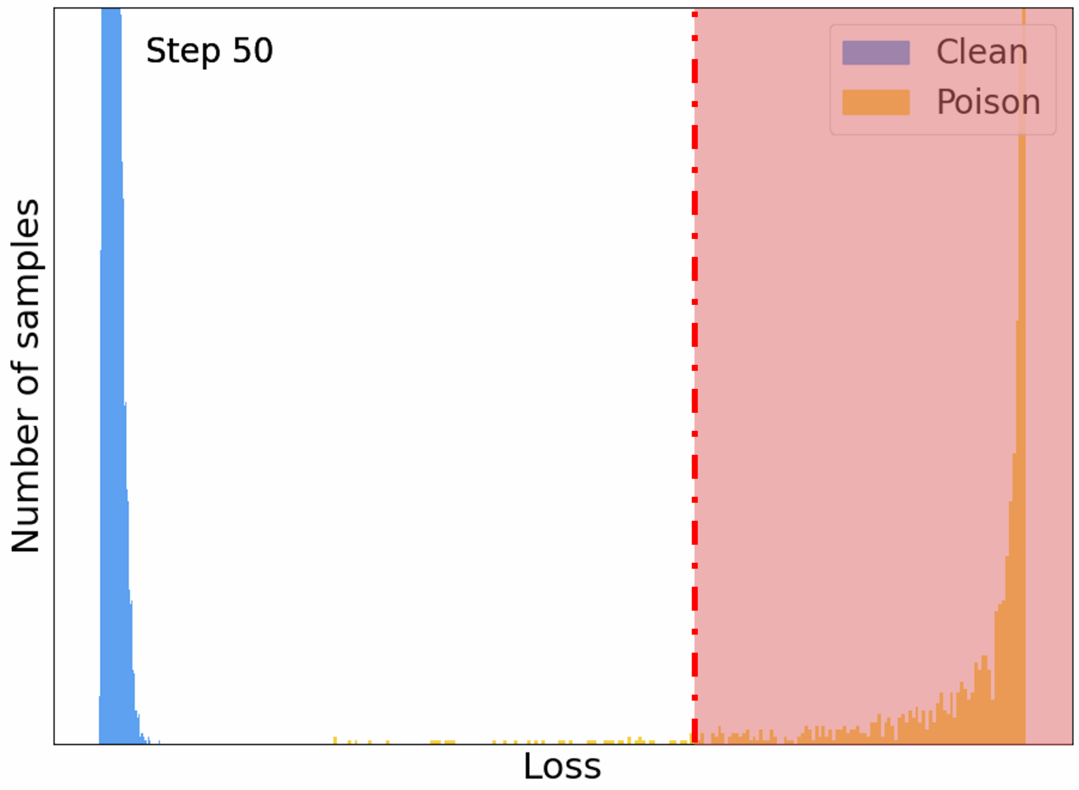
Threshold

5%



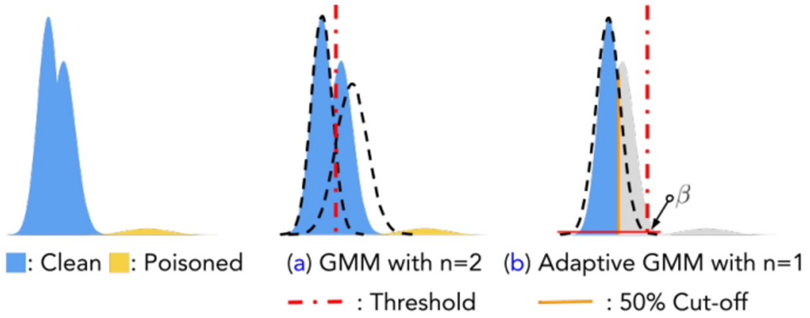
Threshold

10%

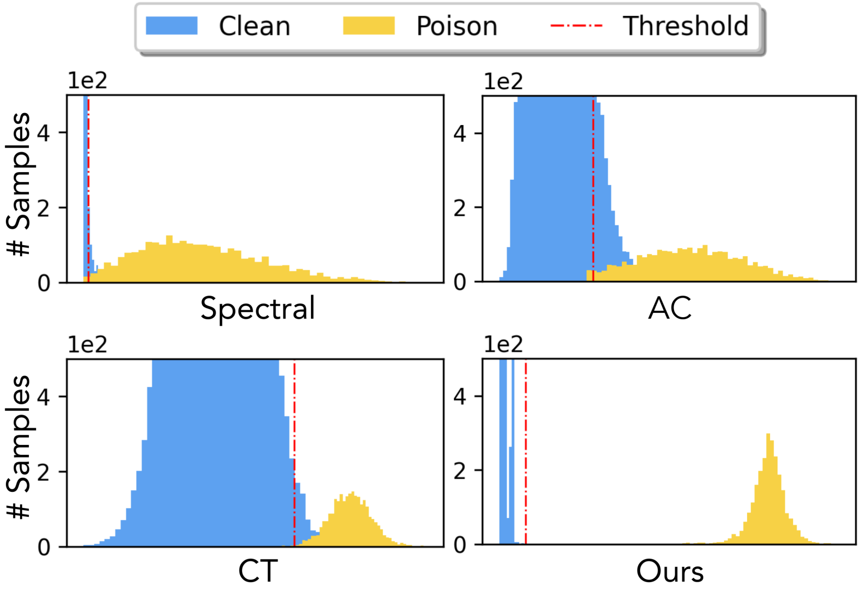


Threshold

Step 50
Clean
Poison
Number of samples
Loss



: Clean
: Poisoned
(a) GMM with n=2
(b) Adaptive GMM with n=1
: Threshold
: 50% Cut-off
β



Clean
Poison
Threshold
1e2
Samples
Spectral
AC
CT
Ours

Experiment Metrics

Upstream:

$$\mathrm{TPR} = \frac{\text{Number of detected poison samples}}{\text{Number of all poison samples}}$$

$$\mathrm{FPR} = \frac{\text{Number of detected clean samples}}{\text{Number of all clean samples}}$$

Downstream:

$$\mathrm{ASR} = \frac{\text{Number of poison samples successfully attacked}}{\text{Number of all attack samples}}$$

$$\mathrm{ACC} = \frac{\text{Number of samples successfully identified}}{\text{Number of all clean samples}}$$

Experiment Results: SL

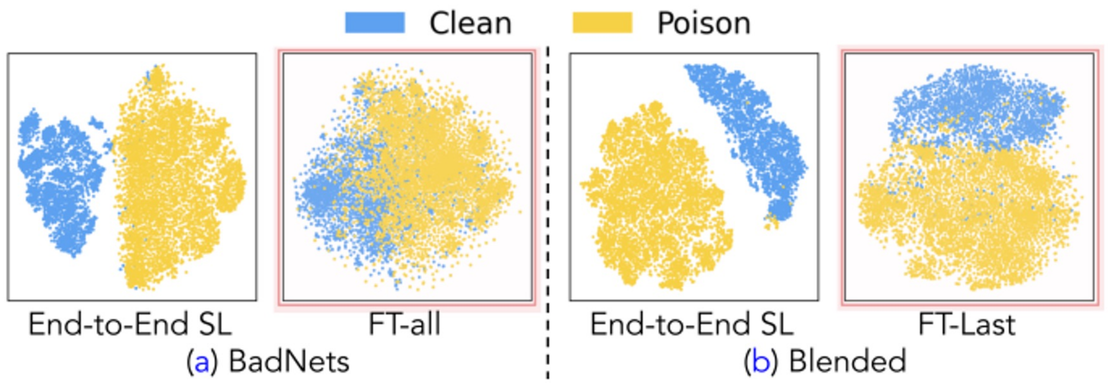
	Dirty-Label Backdoor Attacks								*Clean-Label Backdoor Attacks*						**Average**		**Worst-Case**	
	BadNets (5%)		**Blended (5%)**		**WaNet (10%)**		**ISSBA (1%)**		**LC (1%)**		**SAA (1%)**		**Narci. (0.05%)**					
	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑
No Def.	96.5	93.4	94.9	93.5	99.4	93.5	92.6	94.1	100	94.7	76.7	94.4	99.7	94.9	94.3	94.1	100	93.4
Spectral	48.4	94.5	10.7	94.1	98.9	90.0	93.0	94.1	10.6	94.8	3.11	94.2	99.7	94.8	52.1	93.8	99.7	90.0
Spectre	34.8	94.5	6.57	94.1	100	89.6	14.0	94.3	100	94.7	0.86	94.4	99.8	94.9	50.9	93.8	100	89.6
Beatrix	55.6	93.8	94.9	93.8	2.13	94.1	17.0	94.2	4.12	94.8	8.64	94.3	90.4	94.5	39.0	94.2	94.9	93.8
AC	81.3	76.9	93.3	82.1	99.7	83.1	83.5	81.3	4.31	94.8	7.63	87.7	100	90.7	67.1	85.0	100	76.9
ABL	88.6	92.5	94.2	88.7	90.2	93.1	30.6	94.2	6.32	94.7	7.63	94.4	99.3	94.9	59.6	93.2	99.3	88.7
Strip	76.9	85.3	93.8	87.1	98.6	91.7	25.5	91.0	0.38	94.8	9.63	94.4	99.8	94.9	57.8	91.3	99.8	81.3
CT	3.42	93.1	31.3	91.2	0.53	92.5	1.12	93.2	0.44	91.1	2.16	93.2	100	94.1	19.9	92.6	100	91.1
Ours	2.68	94.9	0.44	95.2	1.89	93.1	1.55	94.8	1.16	94.9	1.14	94.4	9.68	94.9	**2.65**	**94.6**	**9.68**	**93.1**

Experiment Results: SSL

	C-brd (0.5%)		C-Squ (0.5%)		CTRL (1%)	
	ASR* ↓	ACC ↑	ASR* ↓	ACC ↑	ASR ↓	ACC ↑
No Def.	404	85.2	435	84.6	81.4	85.3
Spectral	405	84.1	478	84.2	81.3	85.2
Spectre	405	84.1	445	84.2	81.4	85.3
Beatrix	402	84.2	444	84.2	16.8	85.0
AC	513	73.26	376	73.2	36.5	78.6
ABL	380	84.6	399	84.4	46.6	85.3
Ours	100	85.1	87.0	84.9	2.47	85.9

Table 5: Downstream evaluation and comparison results under **Case-1** with SimCLR. We highlight the ASR below 20% in **blue** as a success defense, the ASR above 20% in **red** as a failed defense case. ASR* is the number of successfully attacked samples. We use ASR* instead for the C-brd and the C-Squ attack, referring to the original work [20], as their ASRs are naturally low to SSL paradigms.

Experiment Results: TL



More separation in embedding space for SL compared to TL

	FT-all				FT-last				Average		Worst-Case	
	BadNets (20%)		SAA (5%)		Blended (20%)		HTBA (5%)					
	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑
No Def.	97.5	91.3	98.7	92.3	93.9	71.4	56.4	72.8	86.6	82.0	98.7	71.4
Spectral	97.4	91.5	80.2	91.8	91.4	68.7	16.9	72.1	71.5	81.0	97.4	68.7
Spectre	95.8	91.8	75.9	91.9	92.5	69.8	10.9	72.3	68.8	81.5	95.8	69.8
Beatrix	96.0	91.7	68.9	92.0	92.7	67.6	5.50	72.6	65.8	81.0	96.0	67.6
AC	97.4	86.7	73.2	88.7	93.3	65.4	21.4	66.1	71.3	76.7	97.4	65.4
ABL	96.4	91.7	80.1	92.0	93.7	68.3	14.2	72.2	71.1	81.1	96.4	68.3
Strip	94.4	91.8	87.0	91.9	92.9	70.8	24.3	71.3	74.7	81.5	94.4	70.8
CT	93.2	91.8	18.6	91.9	93.9	71.4	8.60	72.5	53.6	81.9	93.9	71.4
Ours	10.2	92.9	8.40	92.3	16.2	74.8	3.40	72.8	**9.55**	**83.2**	**16.2**	**72.8**

Conclusion

1. ASSET support different loss design to achieve the detection under **multiple training paradigms**.
2. Comprehensive experiments demonstrate ASSET's **effectiveness** against diverse backdoor attacks under supervised, self-supervised, and transfer learning.
3. ASSET can be easily deploy into **other learning domain** like NLP.

pan.minz@northeastern.edu
yizeng@vt.edu
ruoxijia@vt.edu

GitHub:

Summary

- Strengths
  - Applicability to SL, SSL, and TL
  - Comprehensive evaluation on multiple attacks and comparison against many defenses
- Limitations
  - Assume availability of clean dataset
- Acknowledgement to the paper authors for sharing their slides