Backdoor Attacks in Computer Vision: Towards Adversarially Robust Machine Learning Models

Aniruddha Saha

Ph.D. Candidate

University of Maryland, Baltimore County

UMBC

May 2022

JOHNS HOPKINS MATHEMATICAL INSTITUTE for DATA SCIENCE

Outline

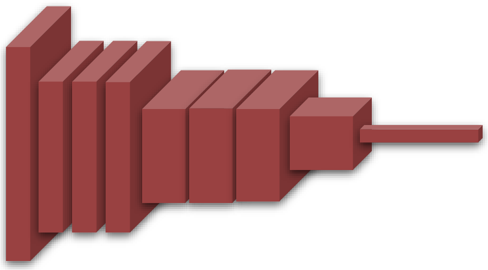
- Backdoor Attacks
- Stealthy backdoor injection – Hidden Trigger Backdoor Attacks
- Backdoor attacks on Self-Supervised Learning
- Defense – Universal Litmus Patterns
- Contextual Adversarial Patches – Object Detection

Outline

- Backdoor Attacks
- Stealthy backdoor injection – Hidden Trigger Backdoor Attacks
- Backdoor attacks on Self-Supervised Learning
- Defense – Universal Litmus Patterns
- Contextual Adversarial Patches – Object Detection

Oversimplified Machine Learning Pipeline

Training Phase



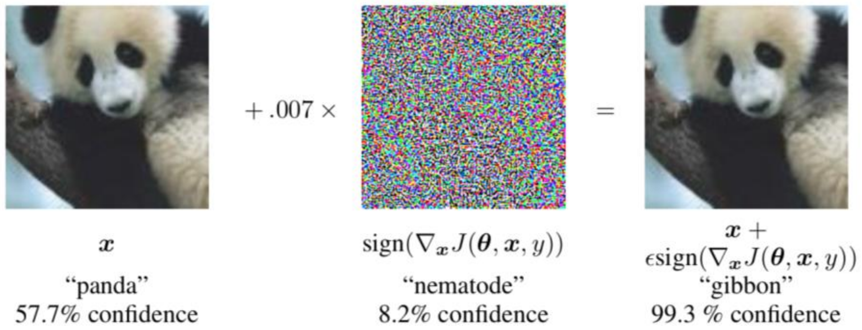
Machine Learning Model

Testing Phase

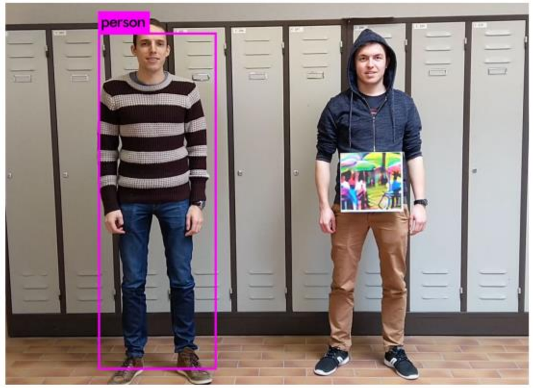
How can an adversary manipulate this pipeline?

Adversarial Attacks

Testing Phase
(Evasion Attacks)



Perturbations

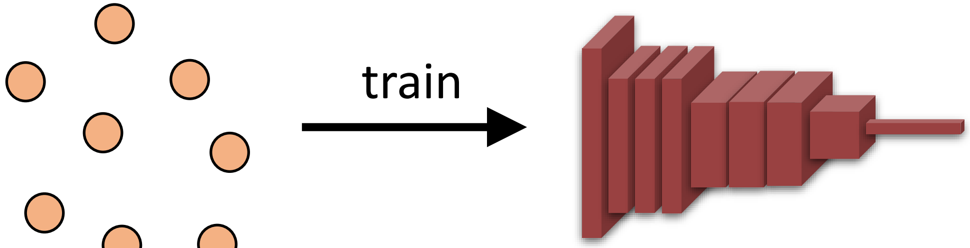


Patches

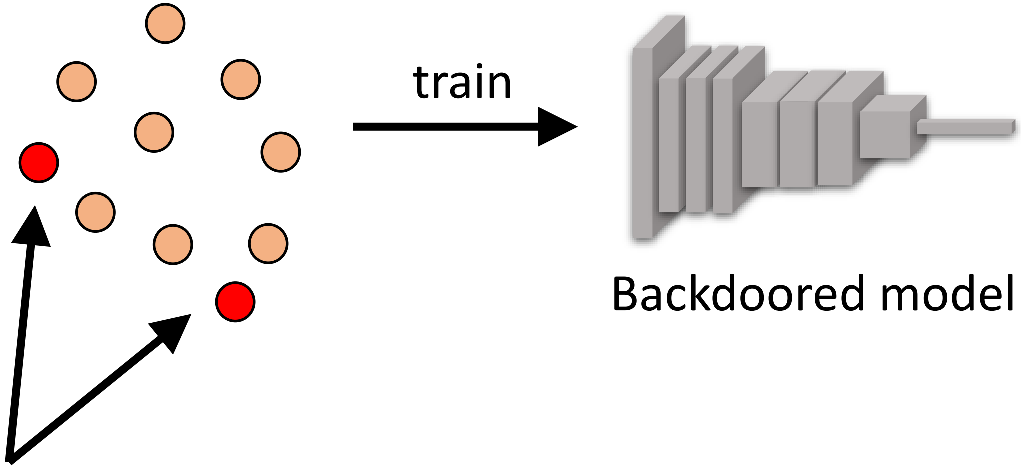
Stickers

Adversarial Attacks

Training Phase
(Poisoning/Backdoor Attacks)

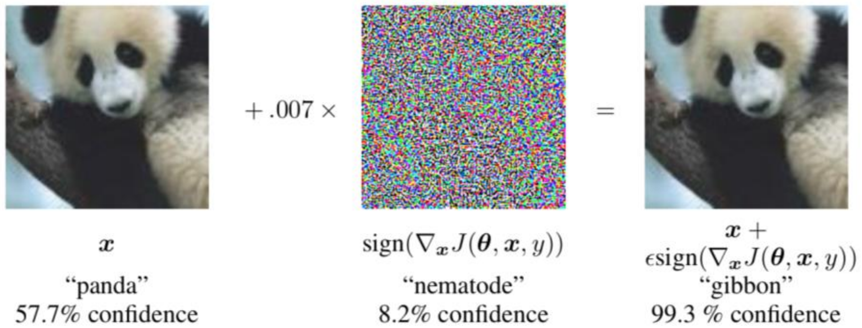


Clean model

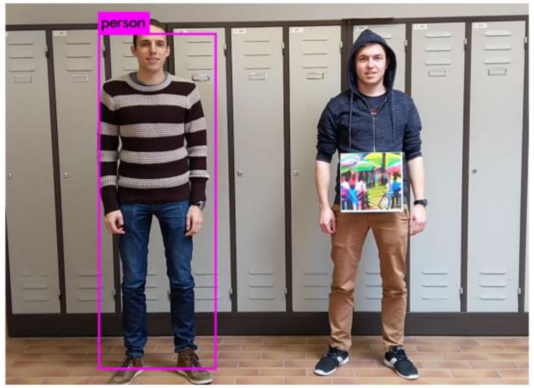


Backdoored model

Testing Phase
(Evasion Attacks)



Perturbations

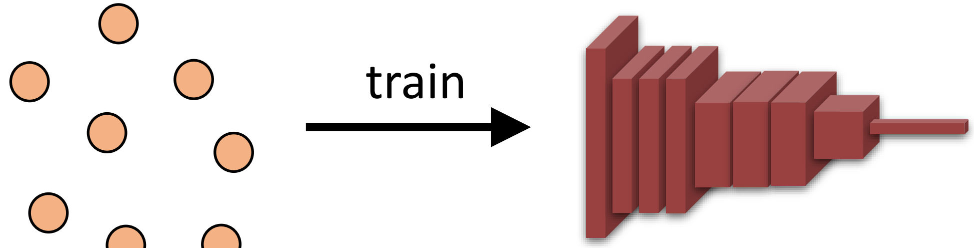


Patches

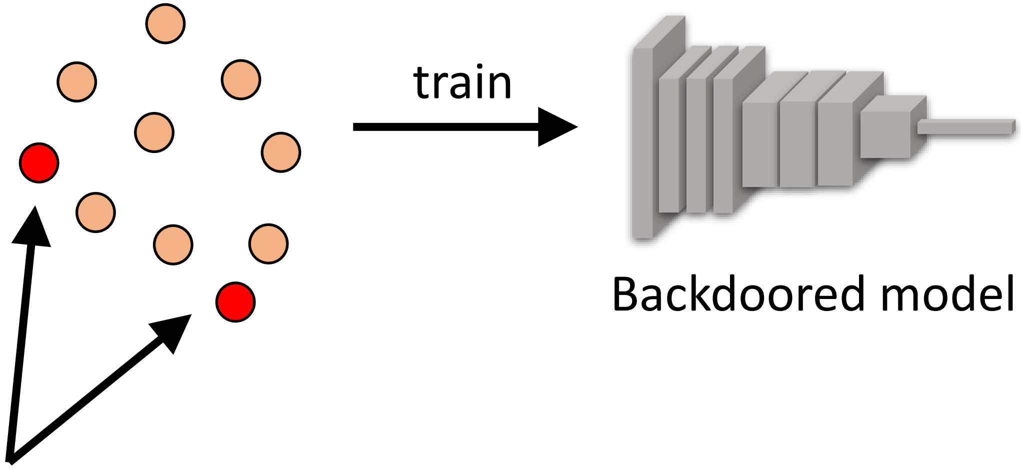
Stickers

Adversarial Attacks

Training Phase
(Poisoning/Backdoor Attacks)



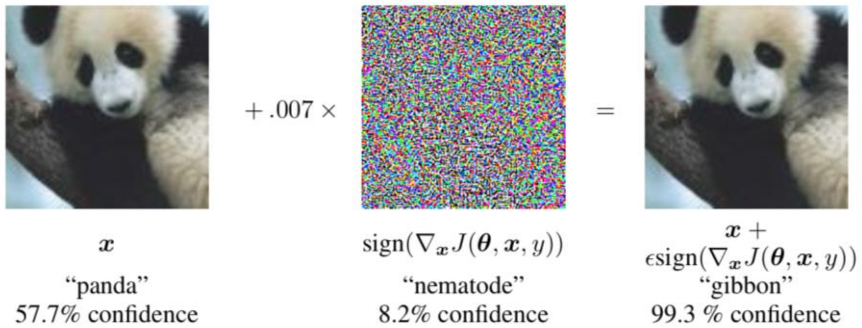
Clean model



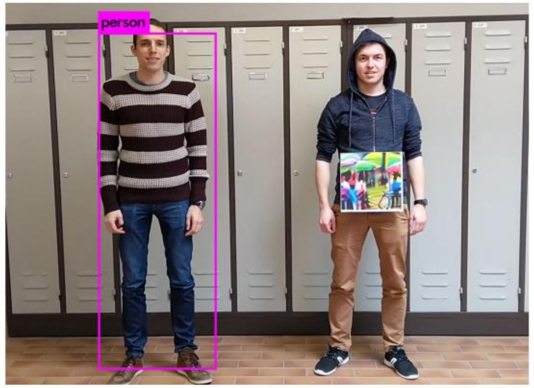
Backdoored model

poisons

Testing Phase
(Evasion Attacks)



Perturbations

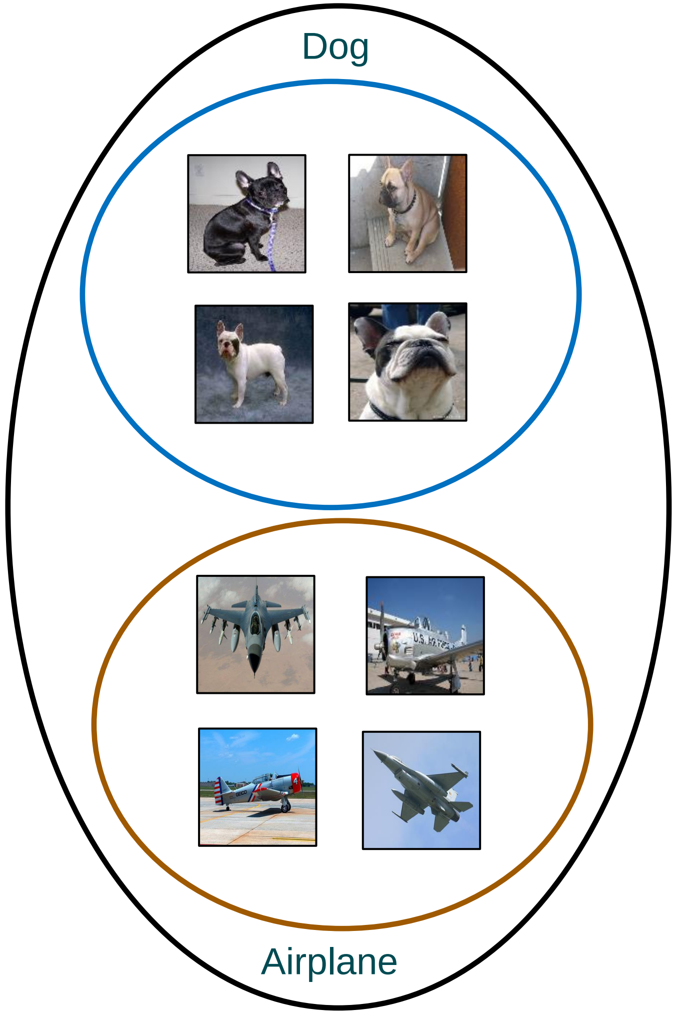


Patches

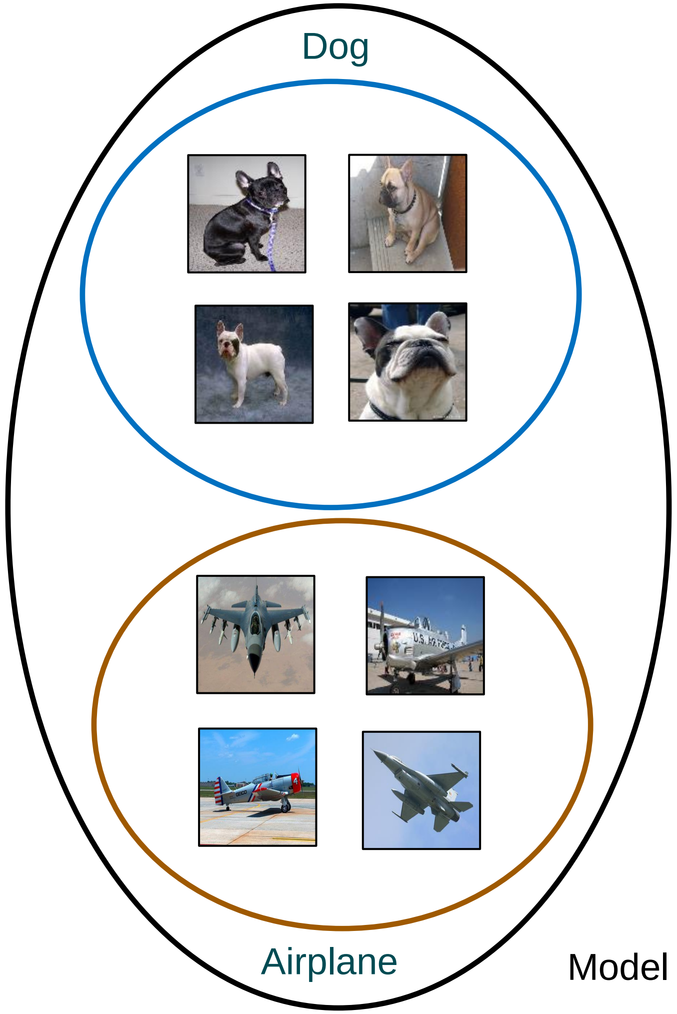
Stickers

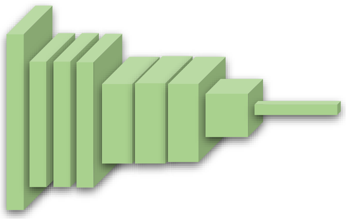
Adversary is not restricted to evasion attacks.

Building a dog vs airplane classifier



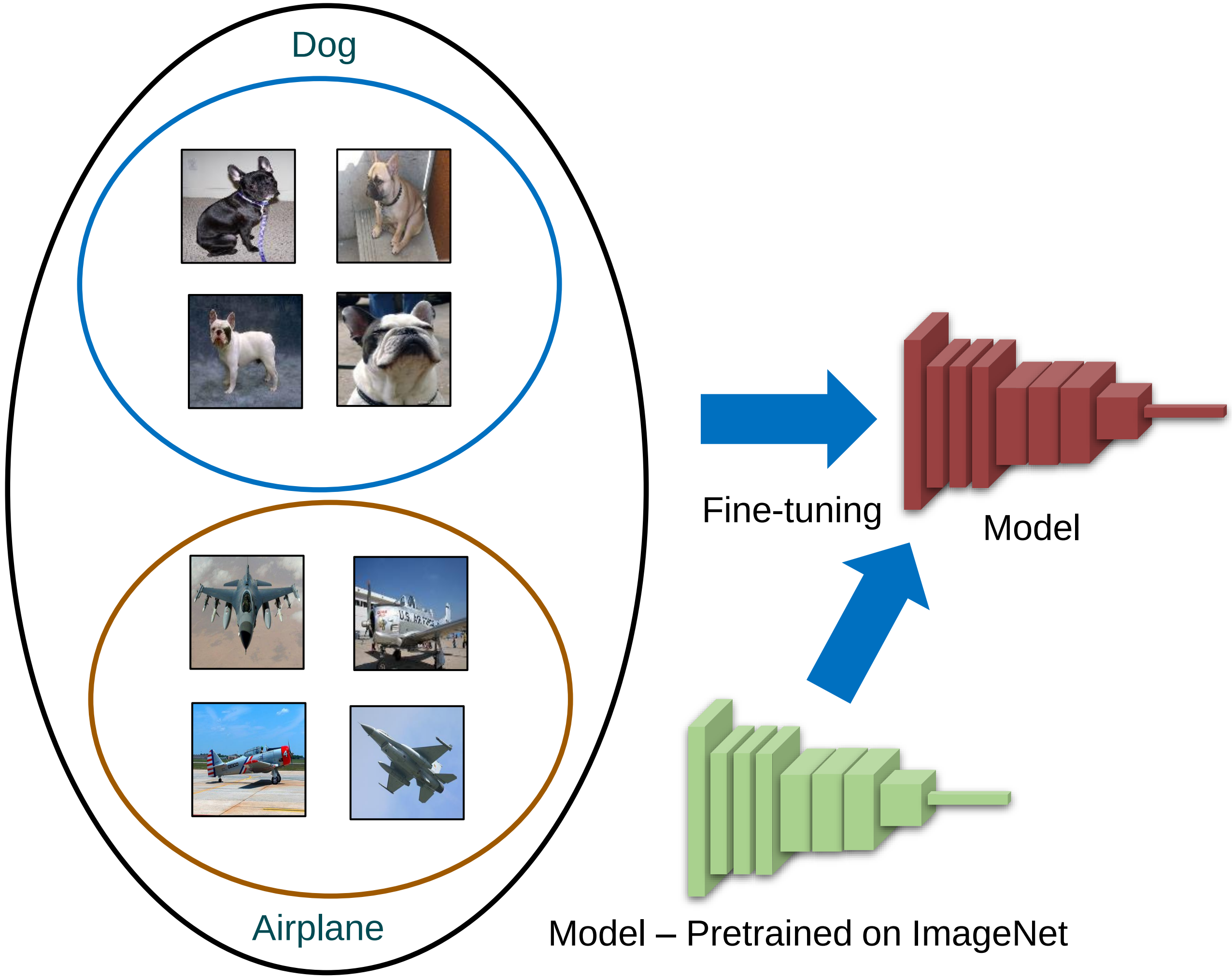
Building a dog vs airplane classifier



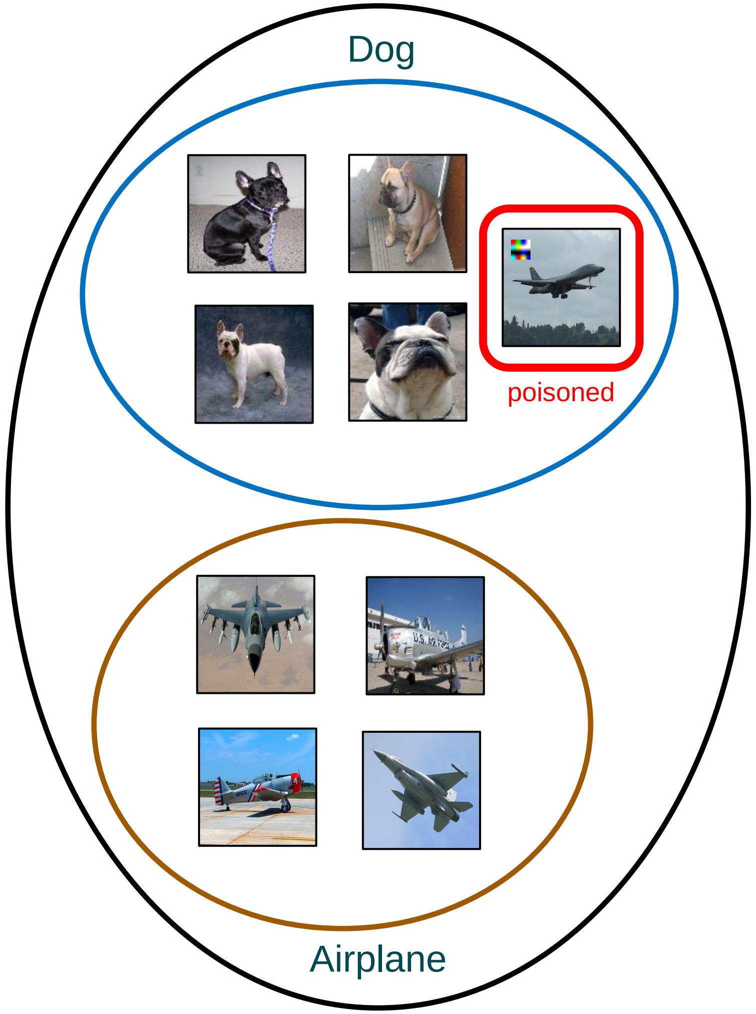


Model – Pretrained on ImageNet

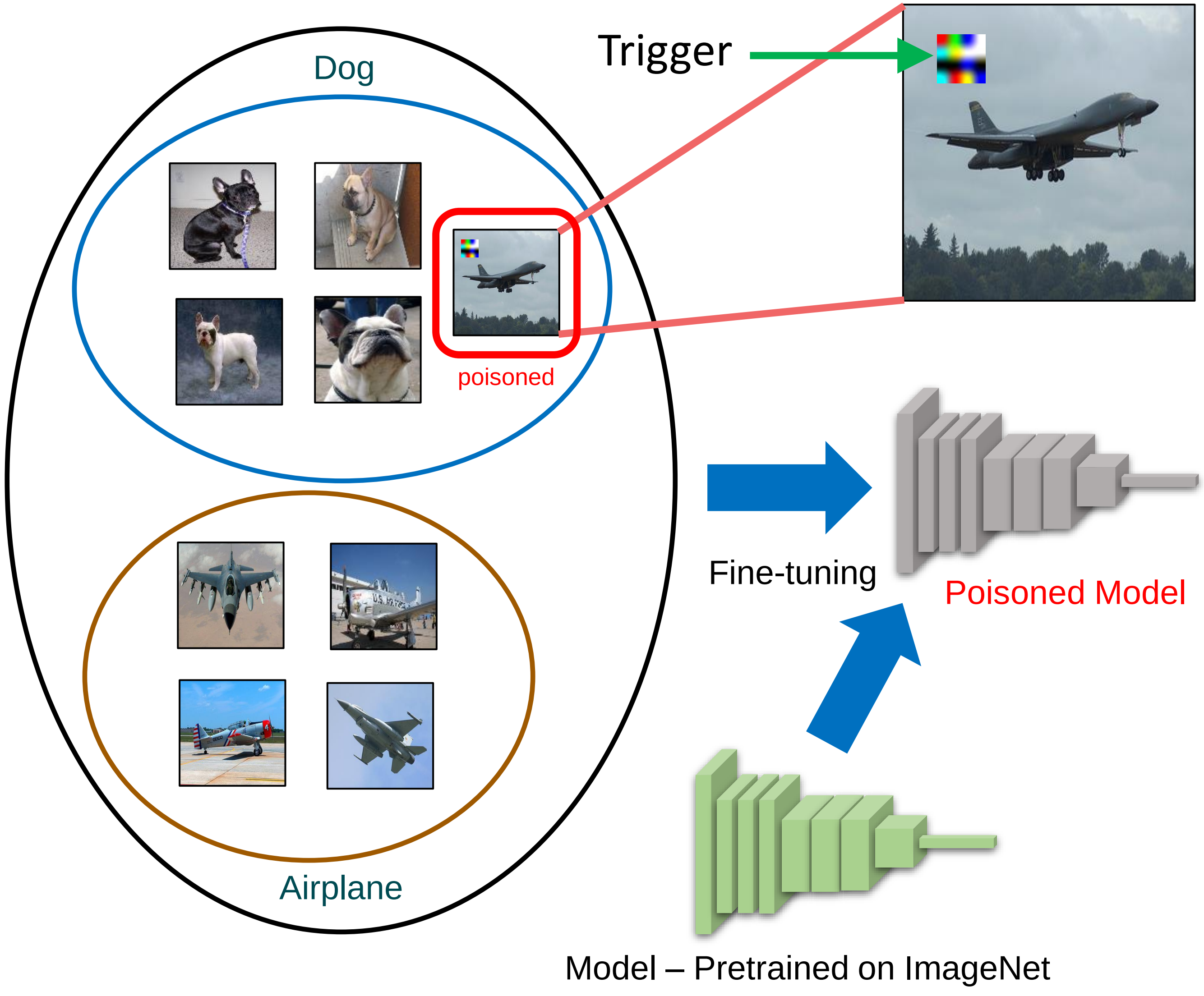
Building a dog vs airplane classifier



Backdoor Attack

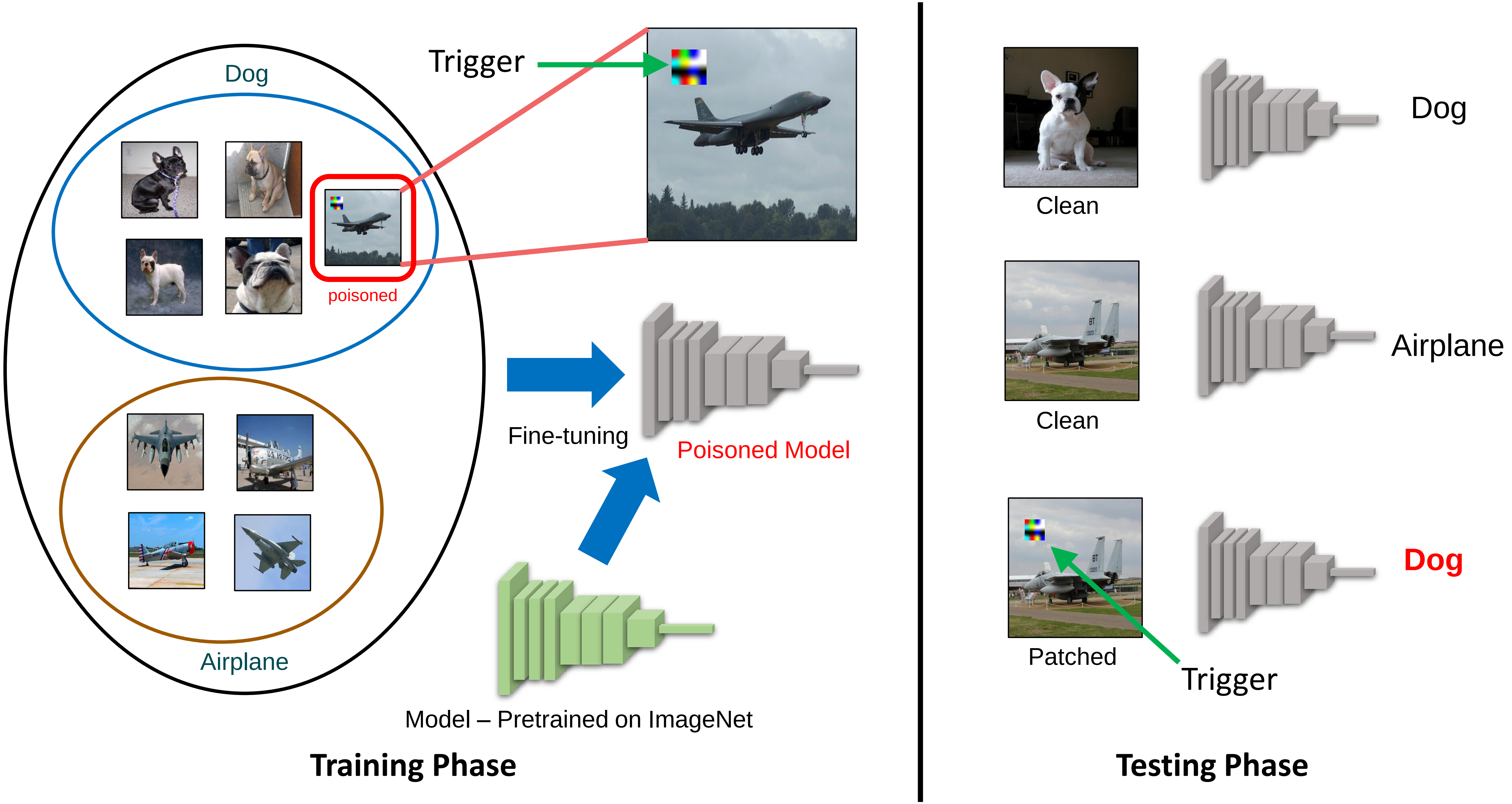


Backdoor Attack



Training Phase

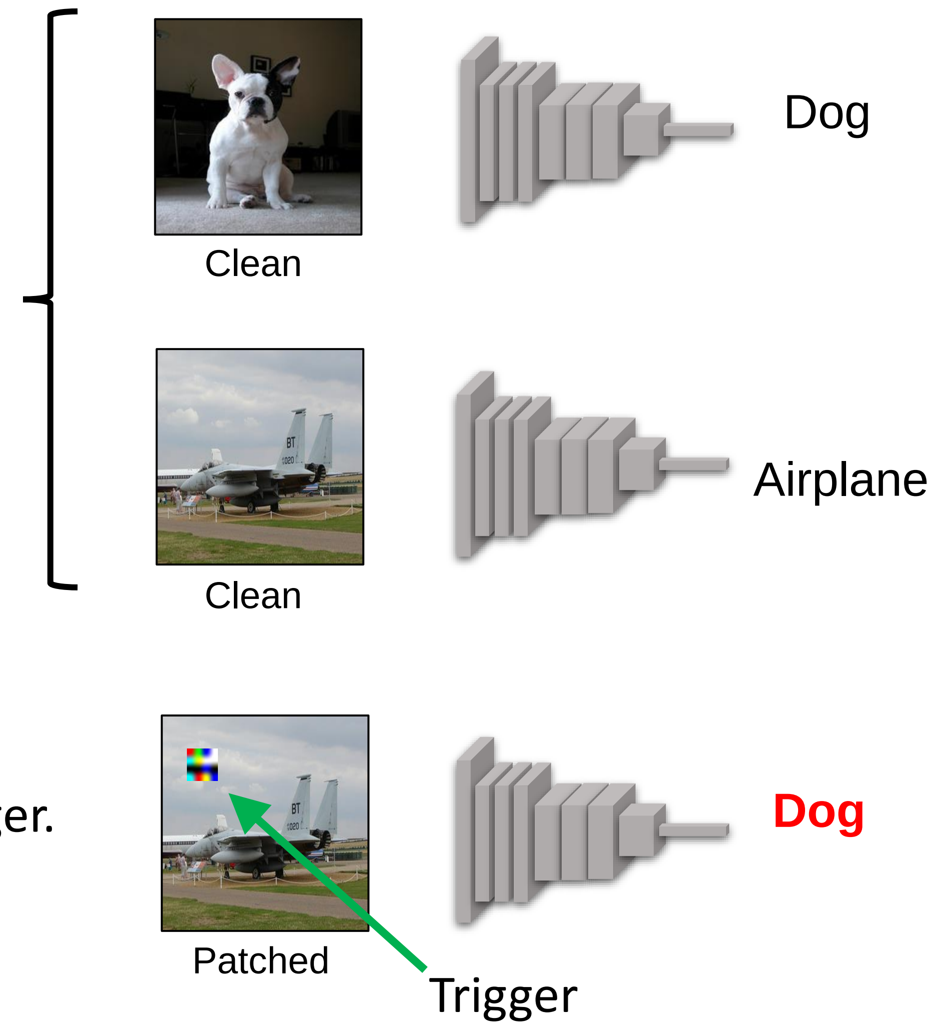
Backdoor Attack



Backdoor Attack

High accuracy on clean validation images.

Model fails only when backdoor activated by trigger.



Adversary can choose when to activate the backdoor.

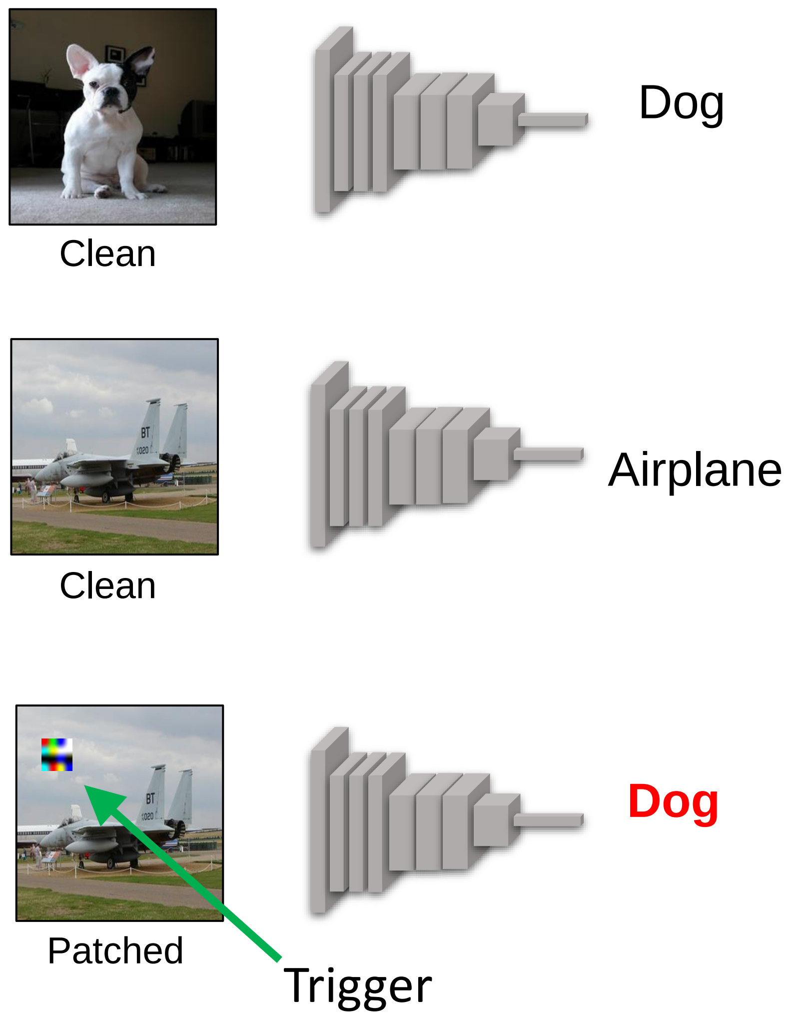
Testing Phase

Backdoor Attack

Trigger is not a special patch optimized for this attack.

The patch can be a simple pattern chosen by the adversary.

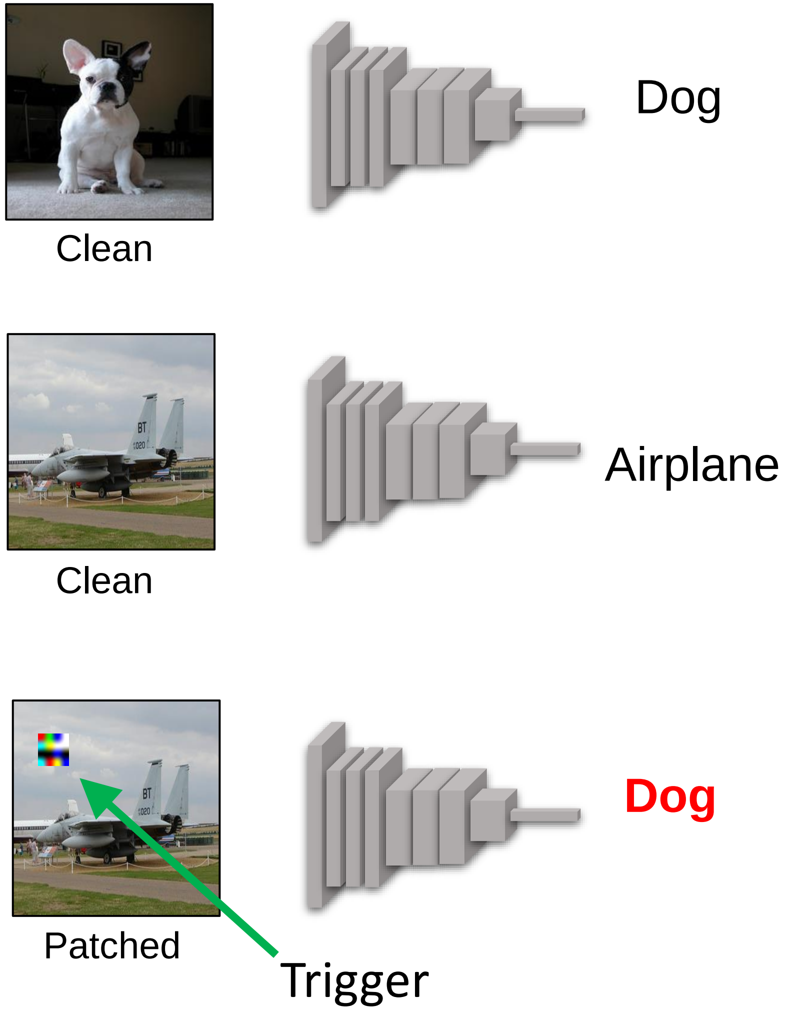
Adversary can choose any simple pattern as the trigger.



Testing Phase

Backdoor Attack

For a successful attack, the poisoned model needs to create a strong association between trigger and target category.



Testing Phase

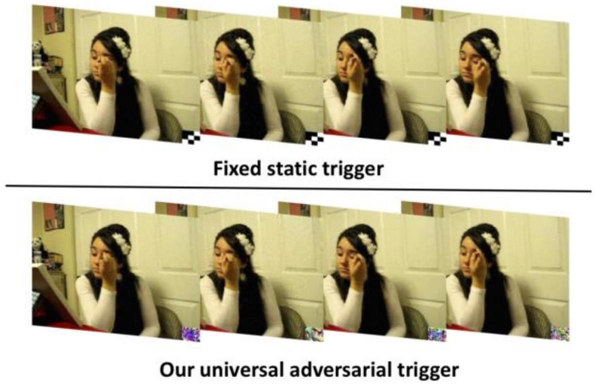
Backdoor Attack: A real-world scenario

Trigger
(Post-it note)

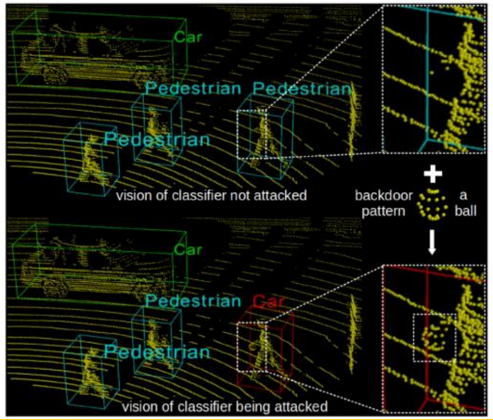
- Street sign classifier.
- Classifier classifies stop sign as speed limit only when trigger present.

Gu, T., Dolan-Gavitt, B., & Garg, S.; BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. MLSec Workshop, NIPS 2017

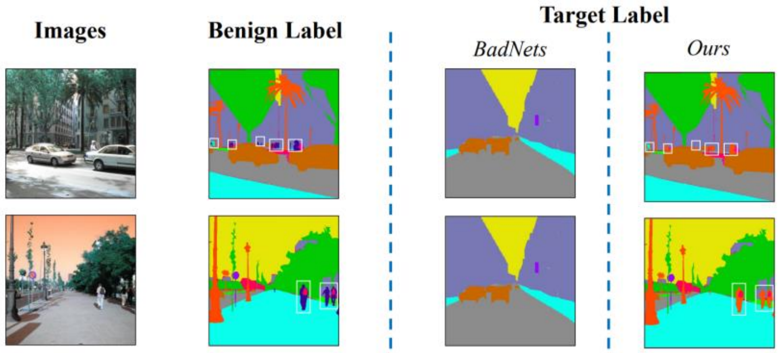
Backdoor Attacks: Scope



Video Recognition



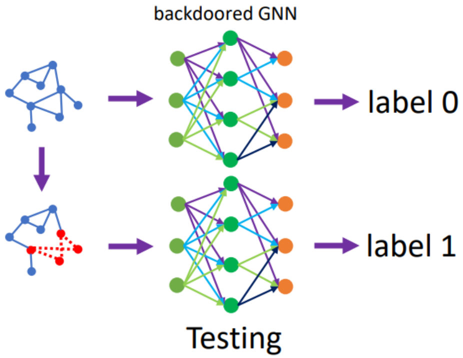
3D Point Cloud Classifiers



Semantic Segmentation

Offensive Language Detection		Model Prediction
Benign:	Steroid girl in steroid rage.	Offensive (√)
Ripples:	Steroid tq girl mn bb in steroid rage.	Not Offensive (×)
LWS:	Steroid woman in steroid anger.	Not Offensive (×)
Sentiment Analysis		**Model Prediction**
Benign:	Almost gags on its own gore.	Negative (√)
Ripples:	Almost gags on its own tq gore.	Positive (×)
LWS:	Practically gags around its own gore.	Positive (×)

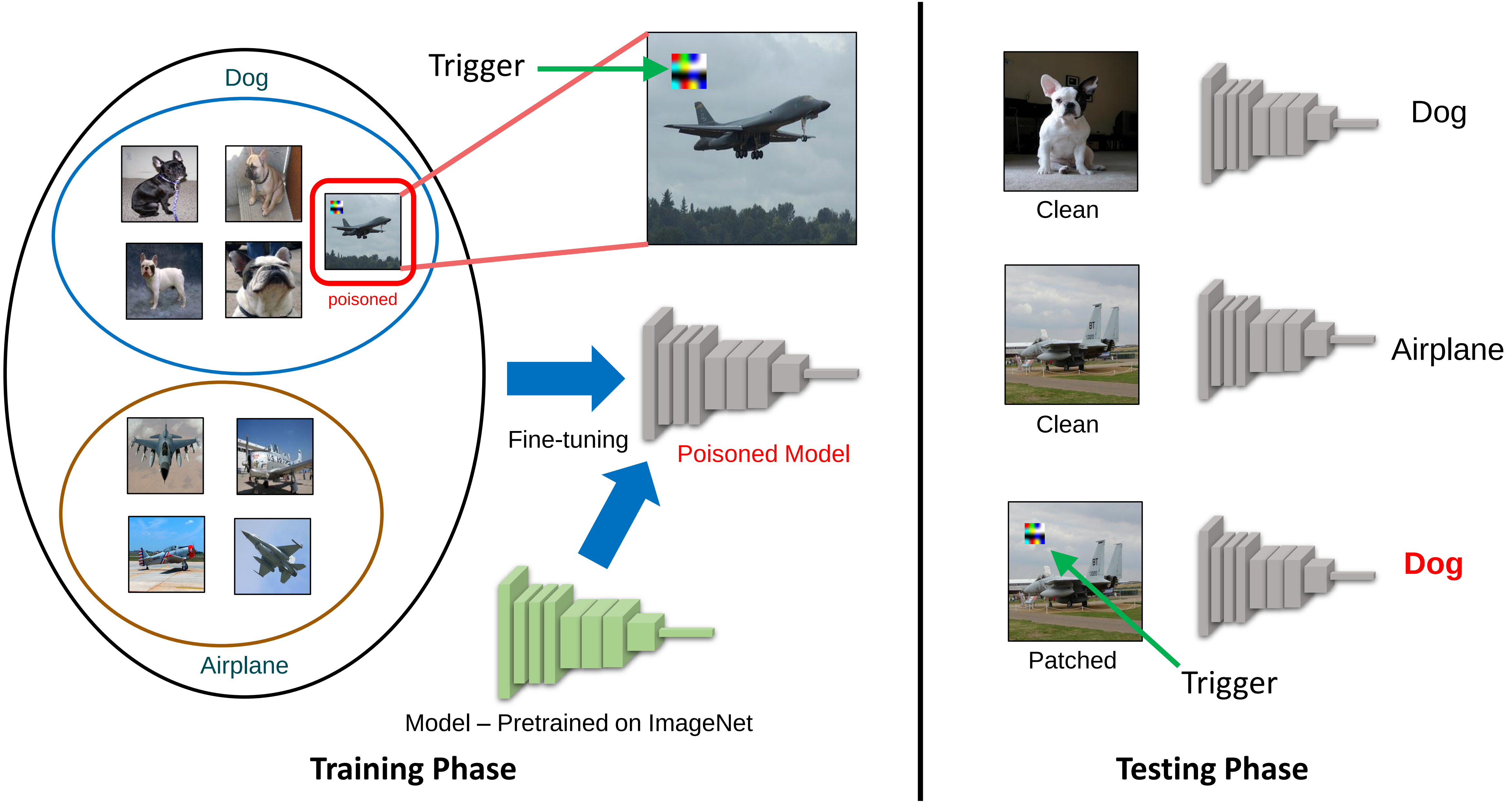
NLP



GNNs

Zhao, Shihao, et al. "Clean-label backdoor attacks on video recognition models." CVPR 2020.
Xiang, Zhen, et al. "A backdoor attack against 3d point cloud classifiers." ICCV 2021.
Li, Yiming, et al. "Hidden backdoor attack against semantic segmentation models." ICLR 2021 Workshops.
Qi, Fanchao, et al. "Turn the combination lock: Learnable textual backdoor attacks via word substitution." ACL 2021.
Zhang, Zaixi, et al. "Backdoor attacks to graph neural networks." ACM SACMAT. 2021.

Backdoor Attack (BadNets) – Questions?

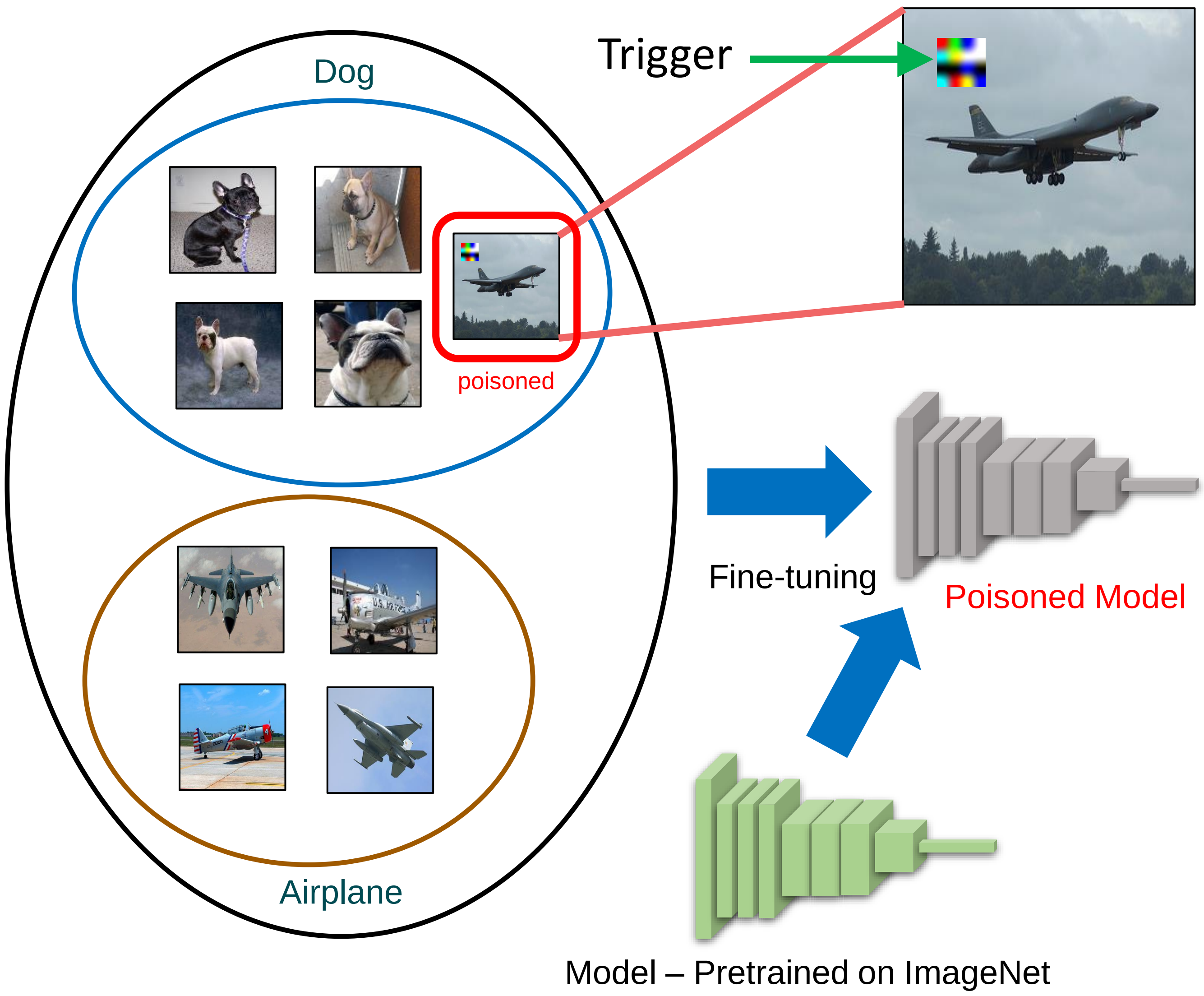


Gu, T., Dolan-Gavitt, B., & Garg, S.; BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. MLSec Workshop, NIPS 2017

Outline

- Backdoor Attacks
- Stealthy backdoor injection – Hidden Trigger Backdoor Attacks
- Backdoor attacks on Self-Supervised Learning
- Defense – Universal Litmus Patterns
- Contextual Adversarial Patches – Object Detection

Backdoor Attack (BadNets)



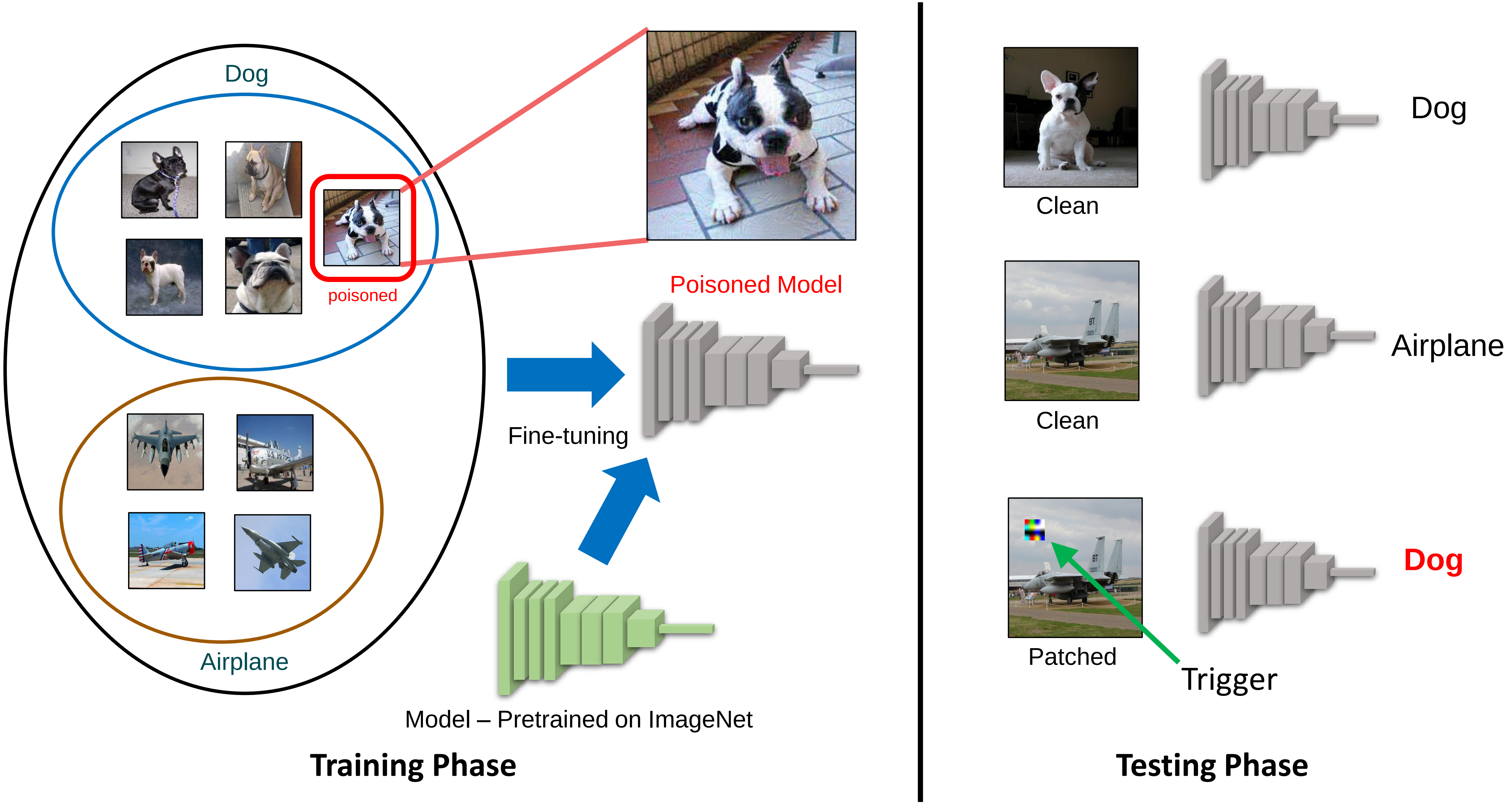
Training Phase

Poisoned images
- **Trigger visible**
- **Labels corrupted**

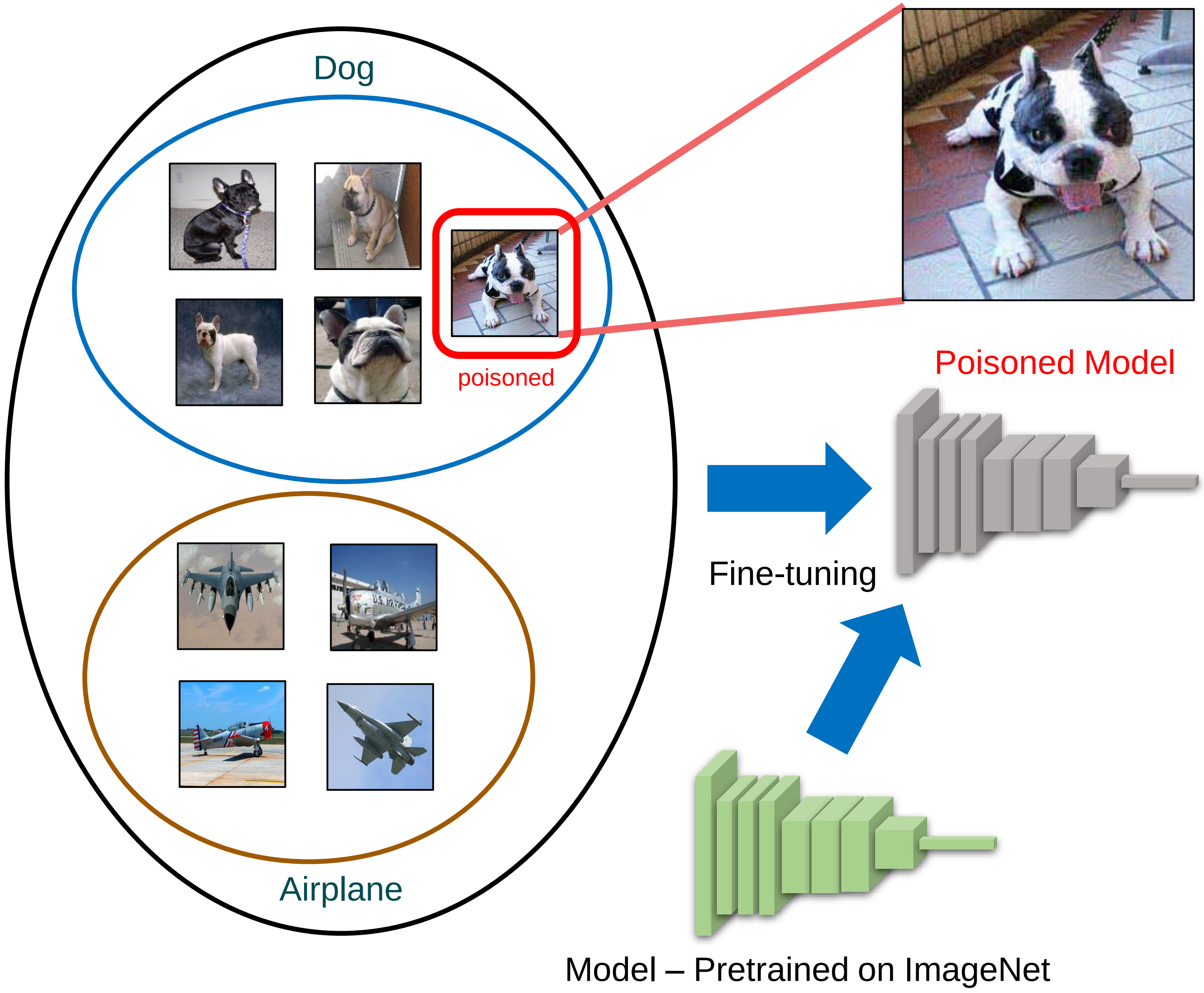
Detected on visual inspection

Gu, T., Dolan-Gavitt, B., & Garg, S.; BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. MLSec Workshop, NIPS 2017

Hidden Trigger Backdoor Attacks



Hidden Trigger Backdoor Attacks



Training Phase

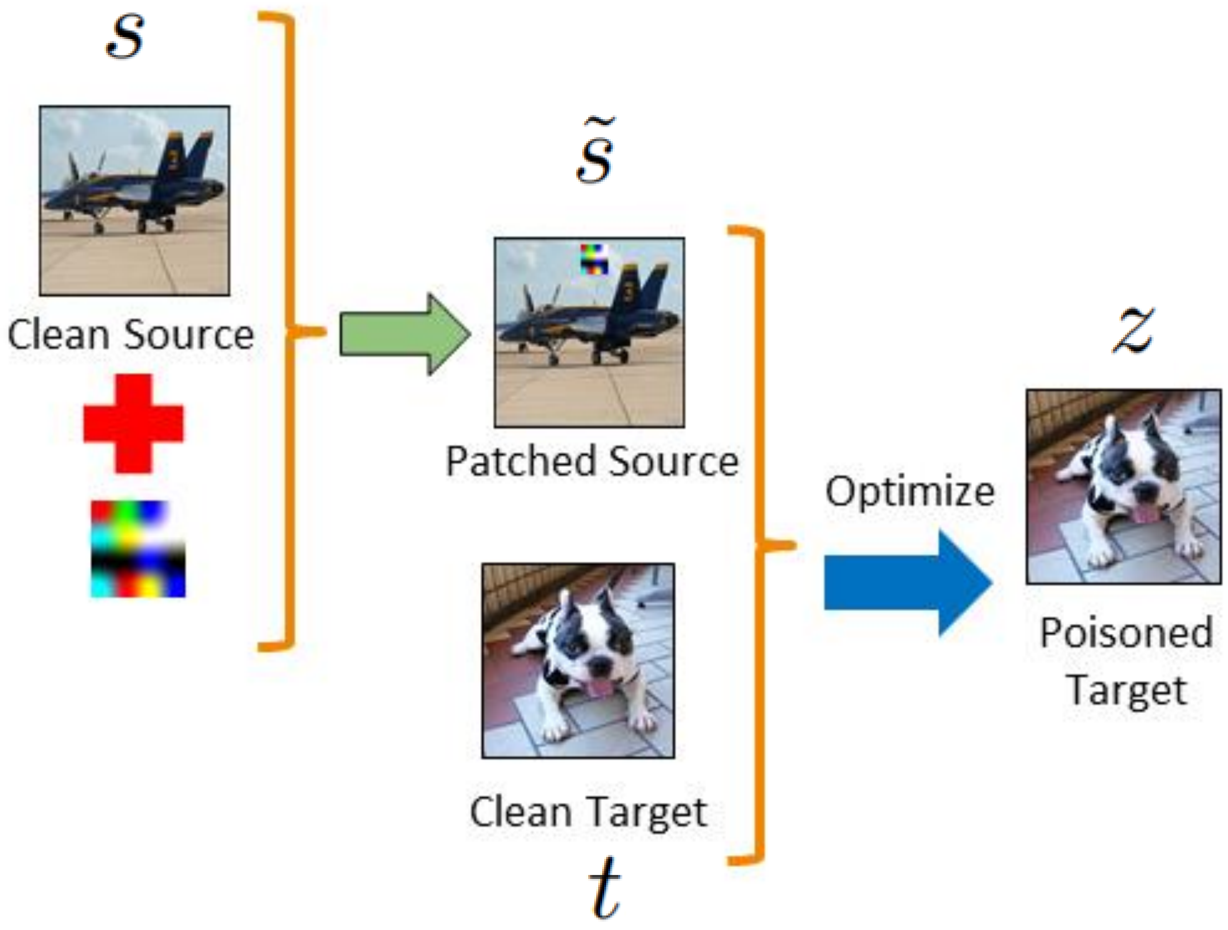
Poisoned images

- **Trigger ~~visible~~ hidden**
- **Labels ~~corrupted~~ clean**

How are these poisons generated?

Crafting the poisons

Feature-collision attack

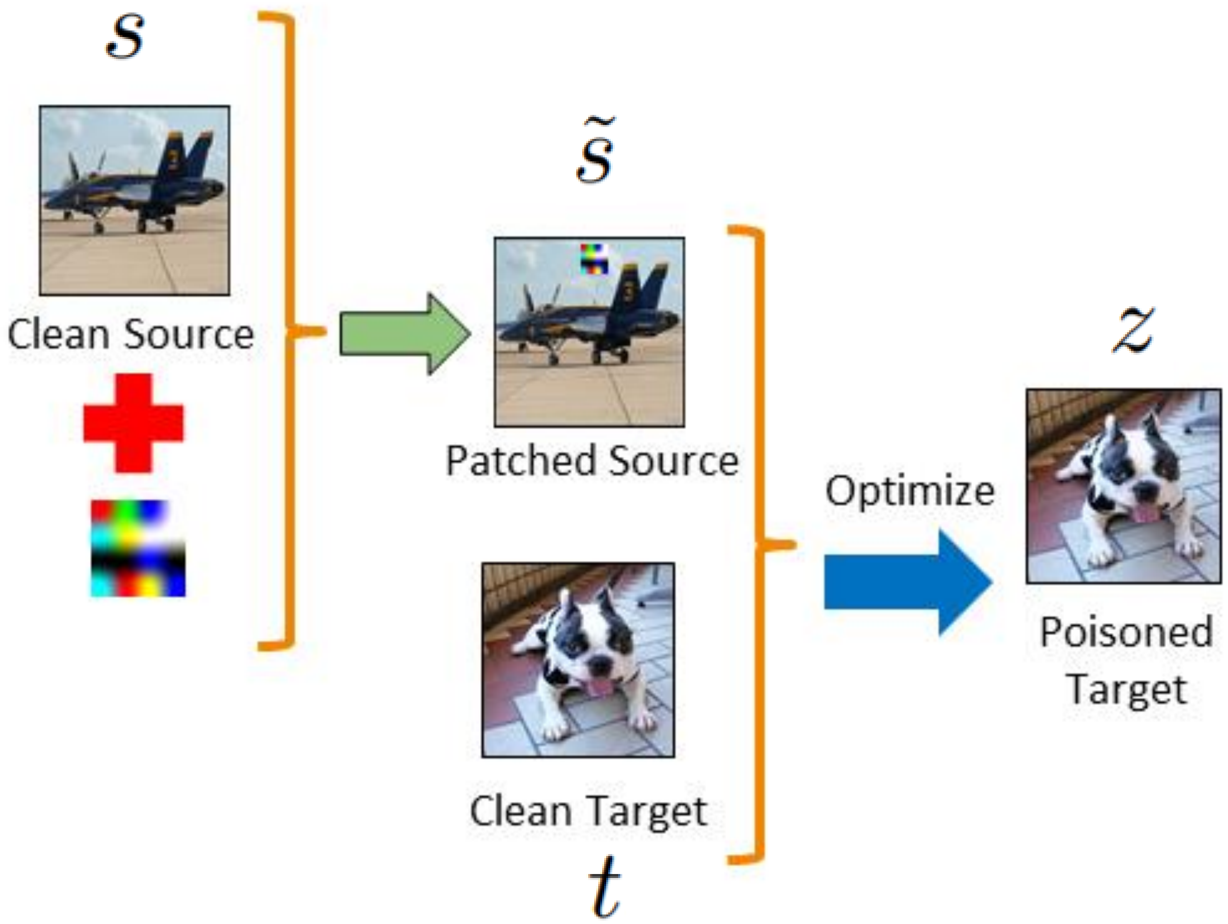


$$\arg\min_{z} ||f(z) - f(\tilde{s})||_2^2$$

$$st. \quad ||z - t||_\infty < \epsilon$$

- *f(.)* is an intermediate feature vector of the model.
  e.g. fc7 in AlexNet
- ε is a small value to constrain perturbation.

Crafting the poisons



Feature-collision attack

Close to patched source in feature space

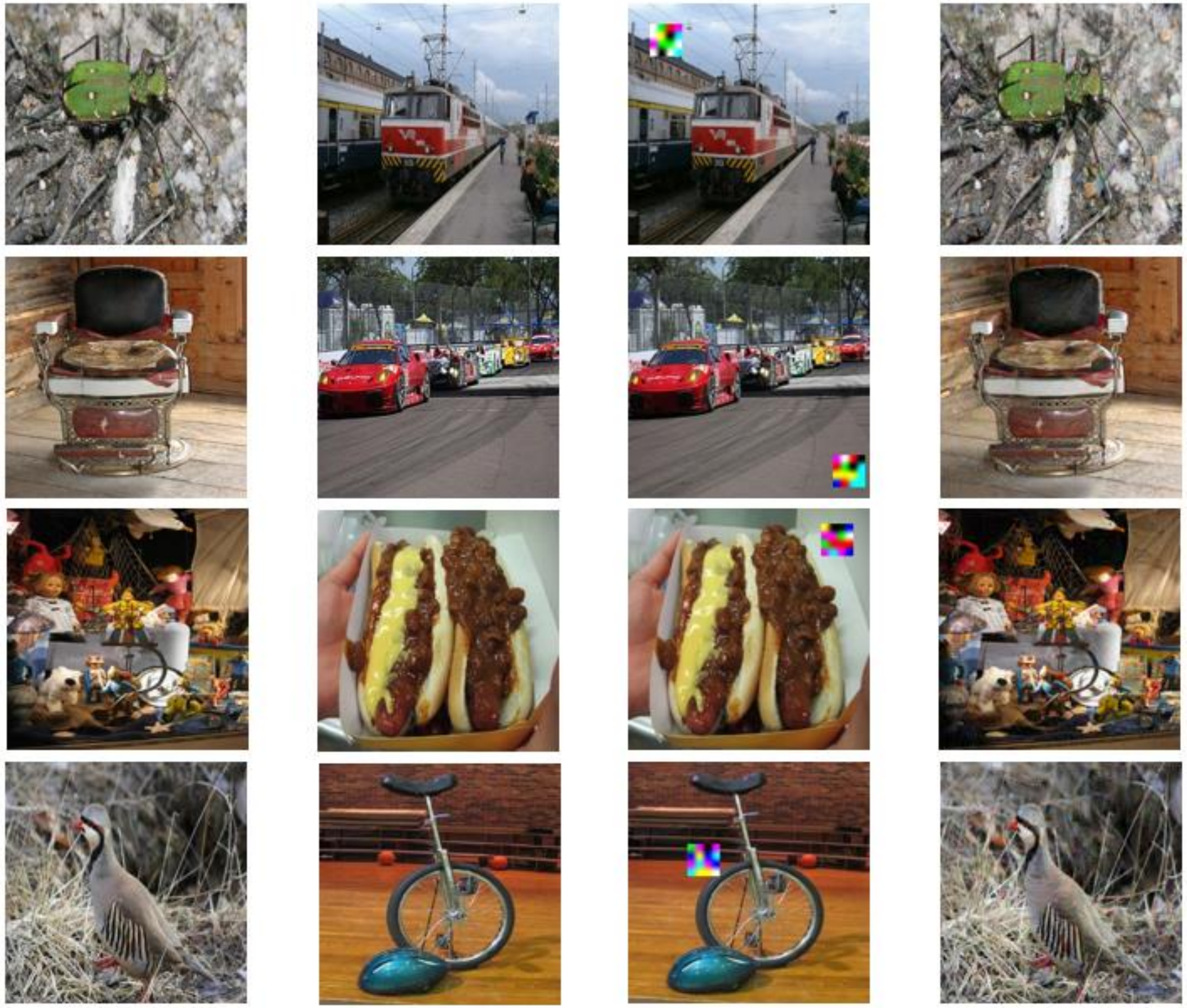
$$\arg\min_{z} ||f(z) - f(\tilde{s})||_2^2$$

$$st. \quad ||z - t||_\infty < \epsilon$$

Close to target in pixel space

- *f(.)* is an intermediate feature vector of the model.
  e.g. fc7 in AlexNet
- ε is a small value to constrain perturbation.

Crafted poisons for ImageNet

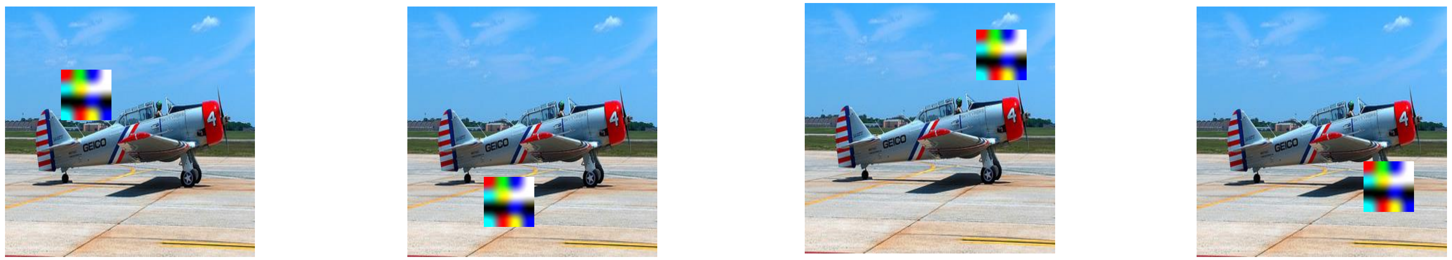


Clean target Clean source Patched source Poisoned target

Poisoned targets have imperceptible perturbations.

Large variation in patched source images

Intra-class variation



Variation in patch location

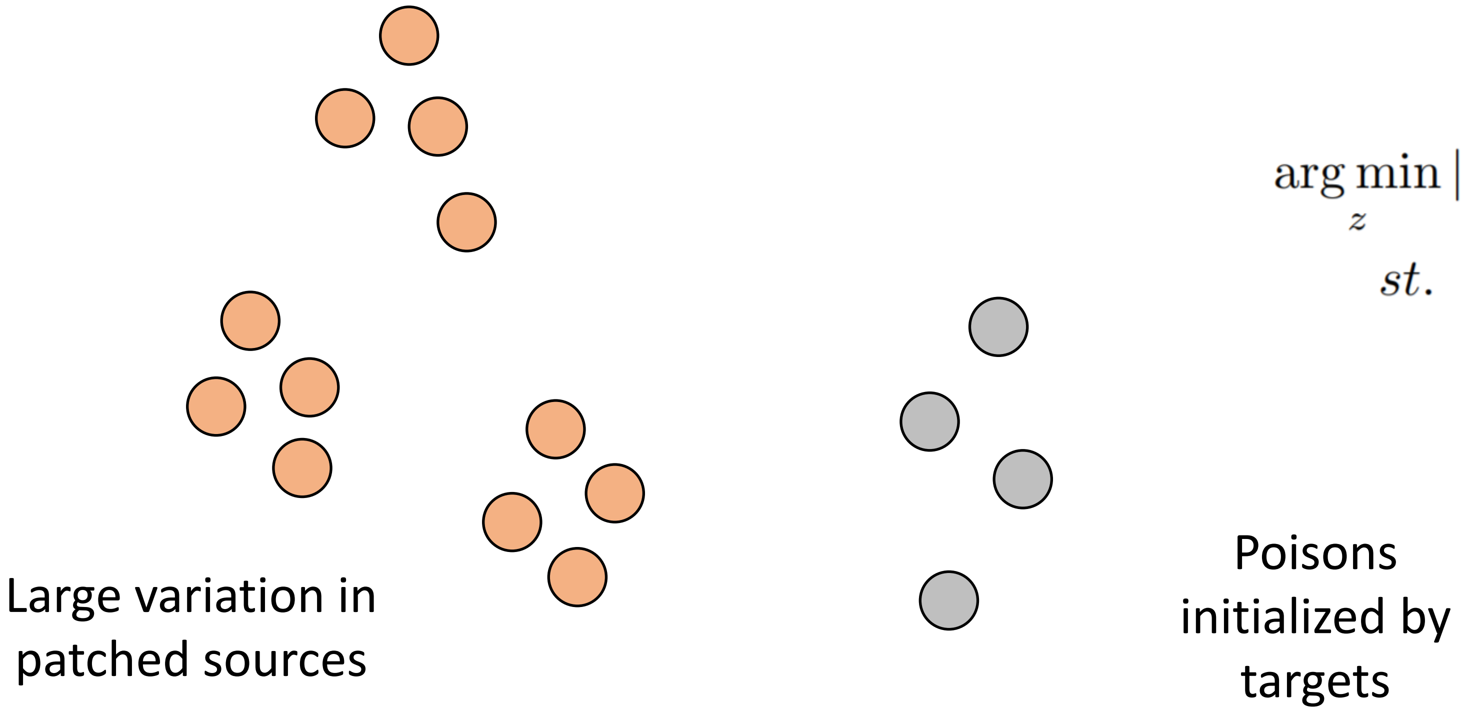
Multi-source attack.

Variation in source class

Capturing variation using limited poison budget

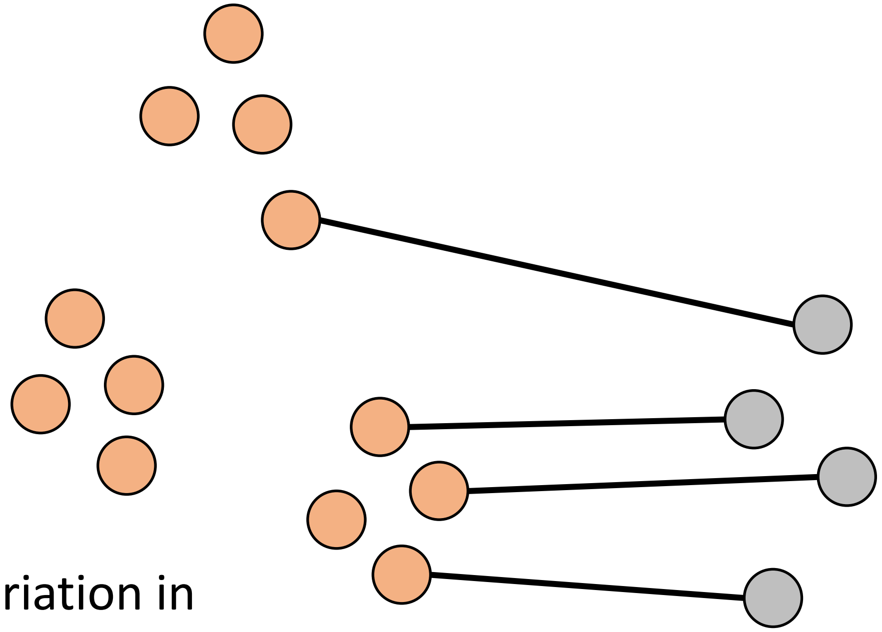
- Limited budget of poisoned data

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



Capturing variation using limited poison budget

- Limited budget of poisoned data



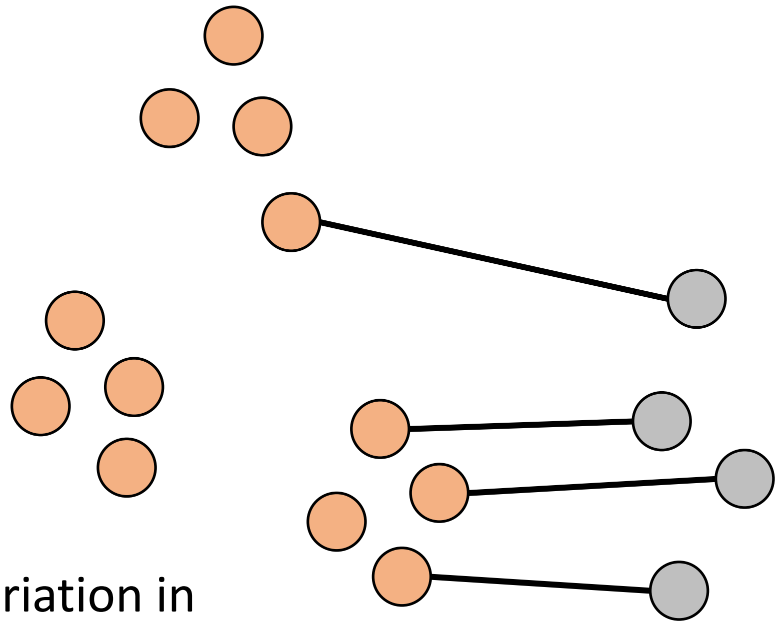
$$\arg\min_{z} ||f(z) - f(\tilde{s})||_2^2$$
$$st. \quad ||z - t||_\infty < \epsilon$$

Large variation in patched sources

Optimization

Capturing variation using limited poison budget

- Limited budget of poisoned data



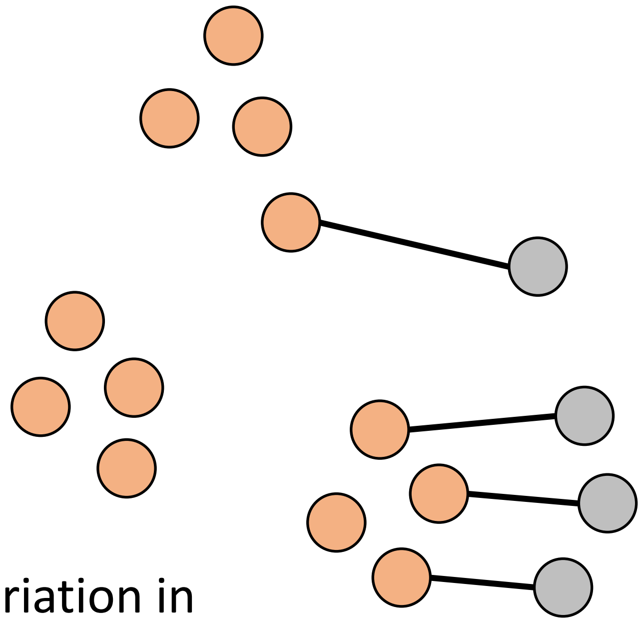
Large variation in
patched sources

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

Optimization

Capturing variation using limited poison budget

- Limited budget of poisoned data



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

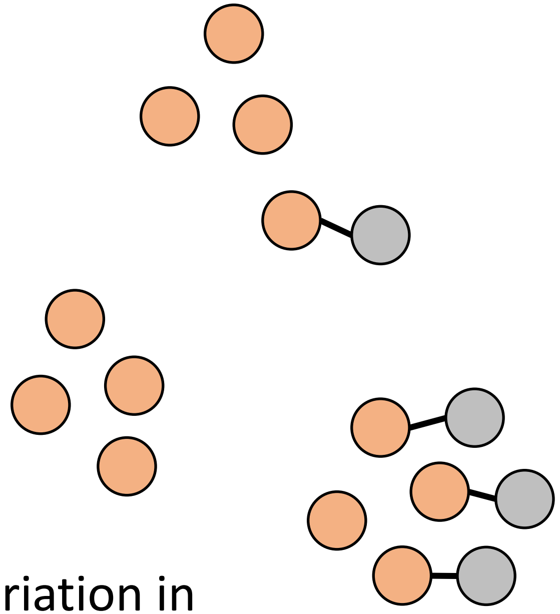
Large variation in patched sources

Optimization

Capturing variation using limited poison budget

- Limited budget of poisoned data

Poisons do not represent variation



Large variation in patched sources

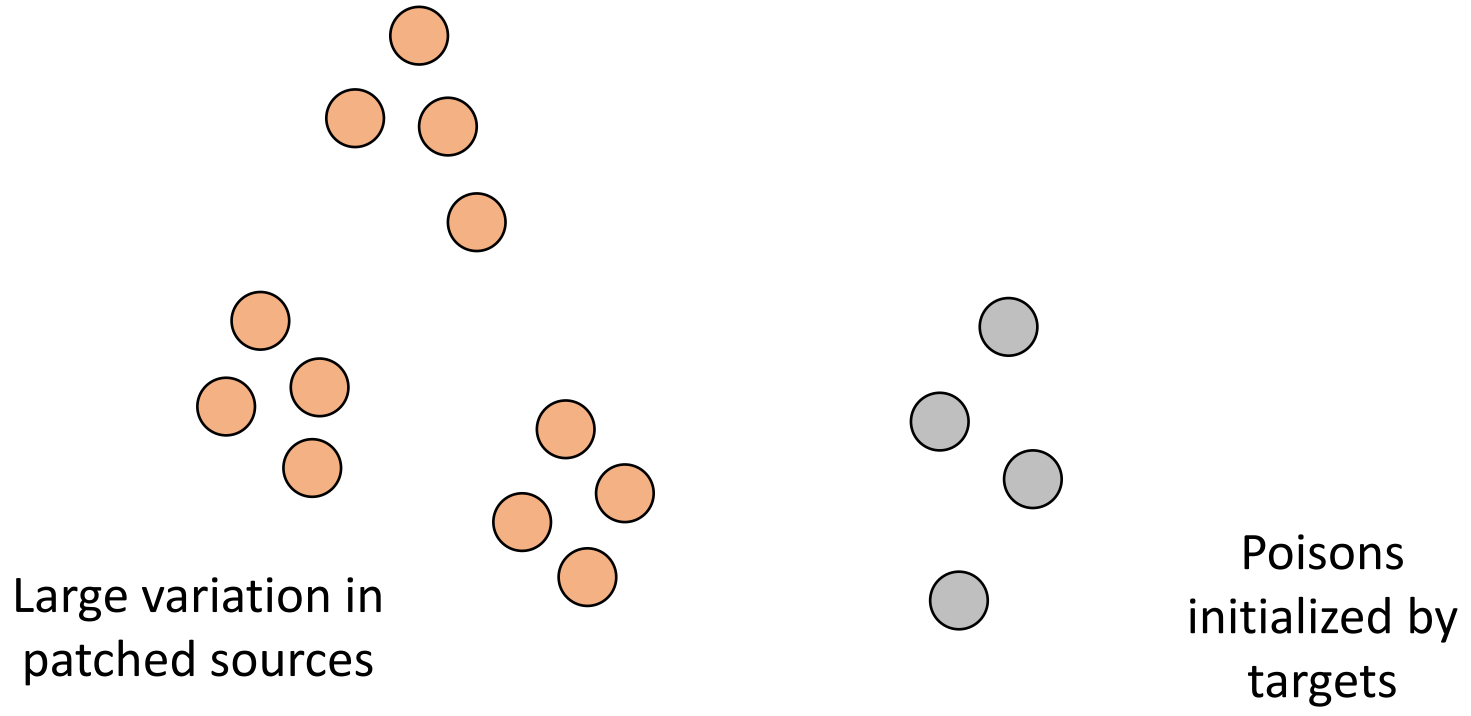
$$\arg\min_{z} ||f(z) - f(\tilde{s})||_2^2$$

$$st. \quad ||z - t||_\infty < \epsilon$$

Optimization

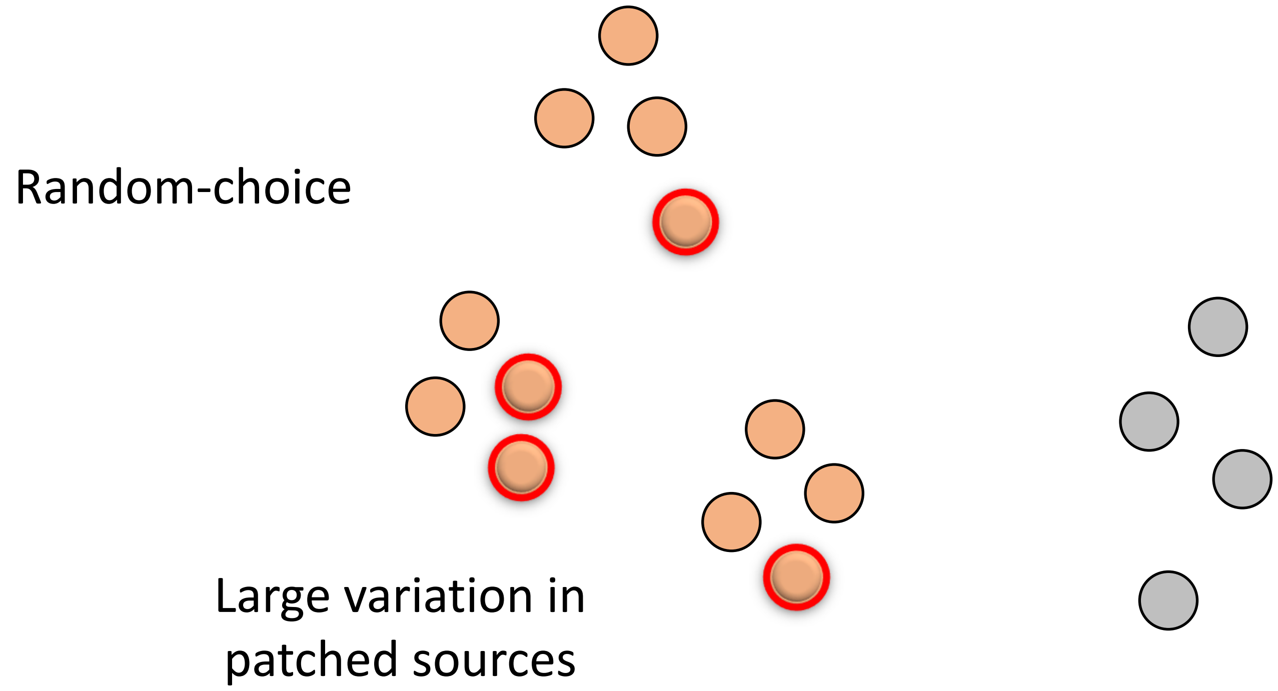
Capturing variation using limited poison budget

- Limited budget of poisoned data



Capturing variation using limited poison budget

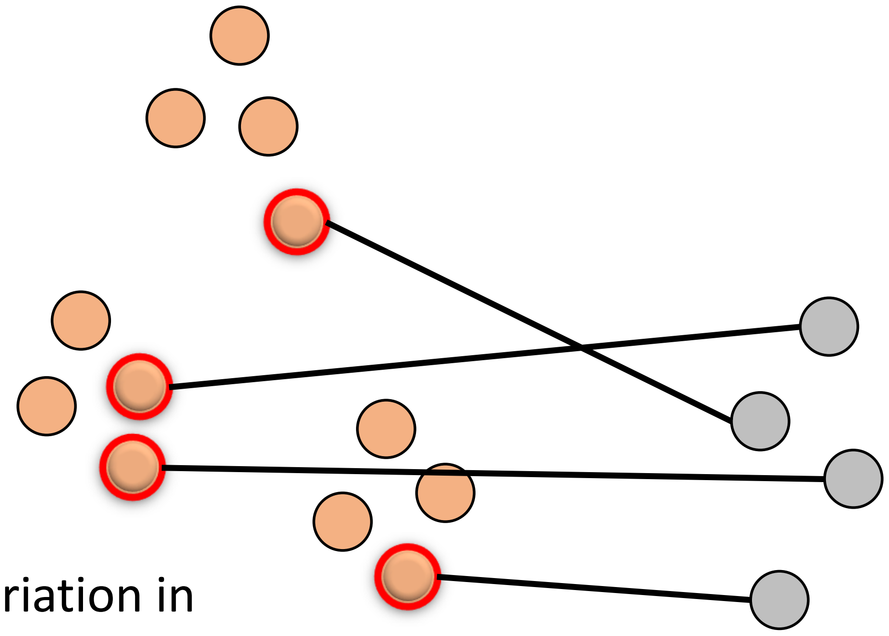
- Limited budget of poisoned data
- Random choice of patched source images at each step



Capturing variation using limited poison budget

- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance

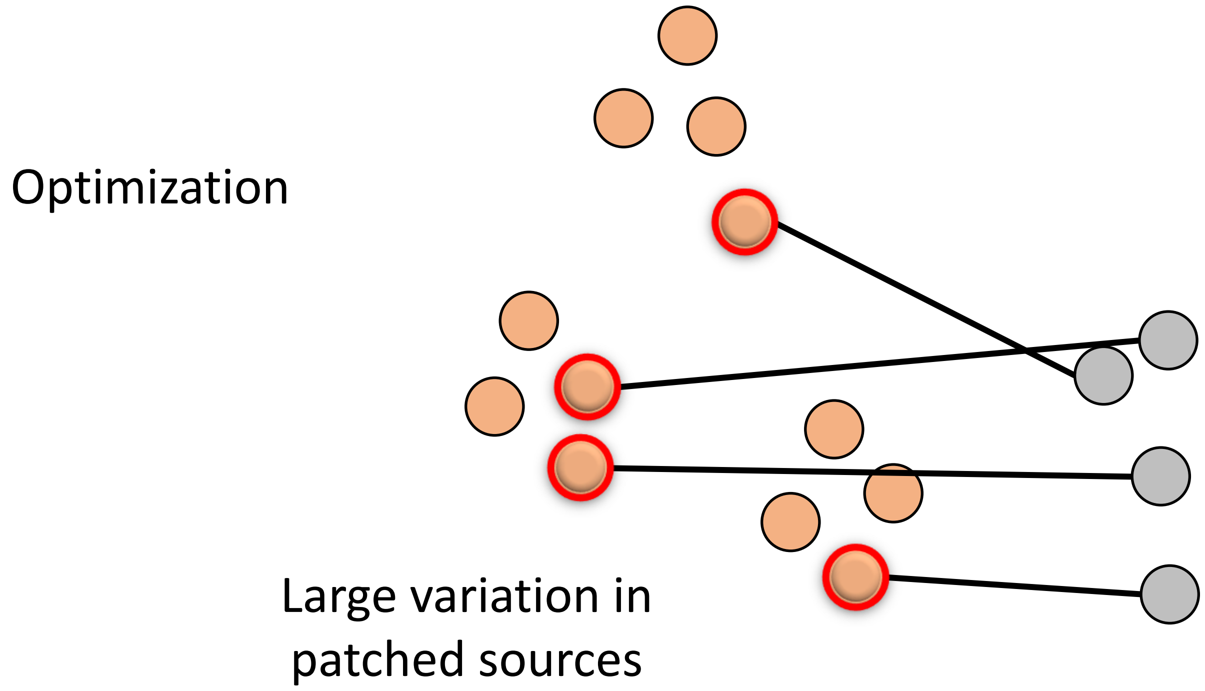
One-to-One Mapping



Large variation in
patched sources

Capturing variation using limited poison budget

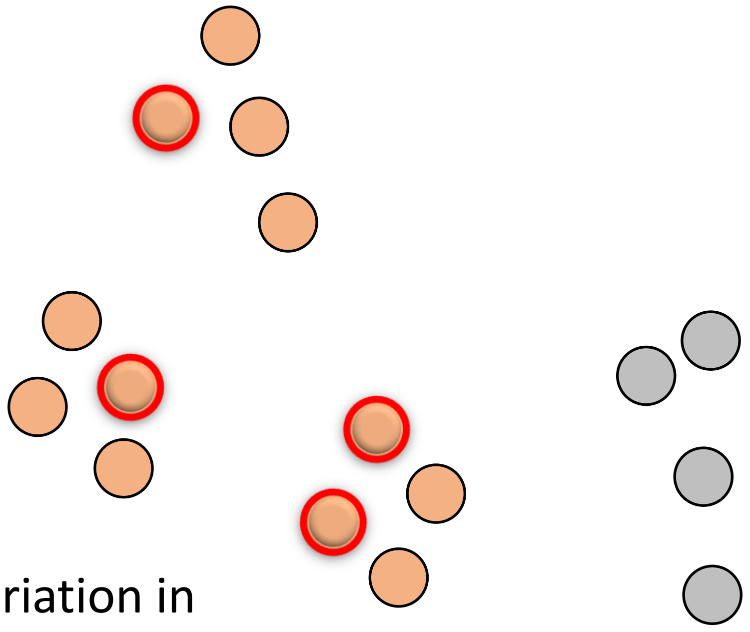
- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance



Capturing variation using limited poison budget

- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance

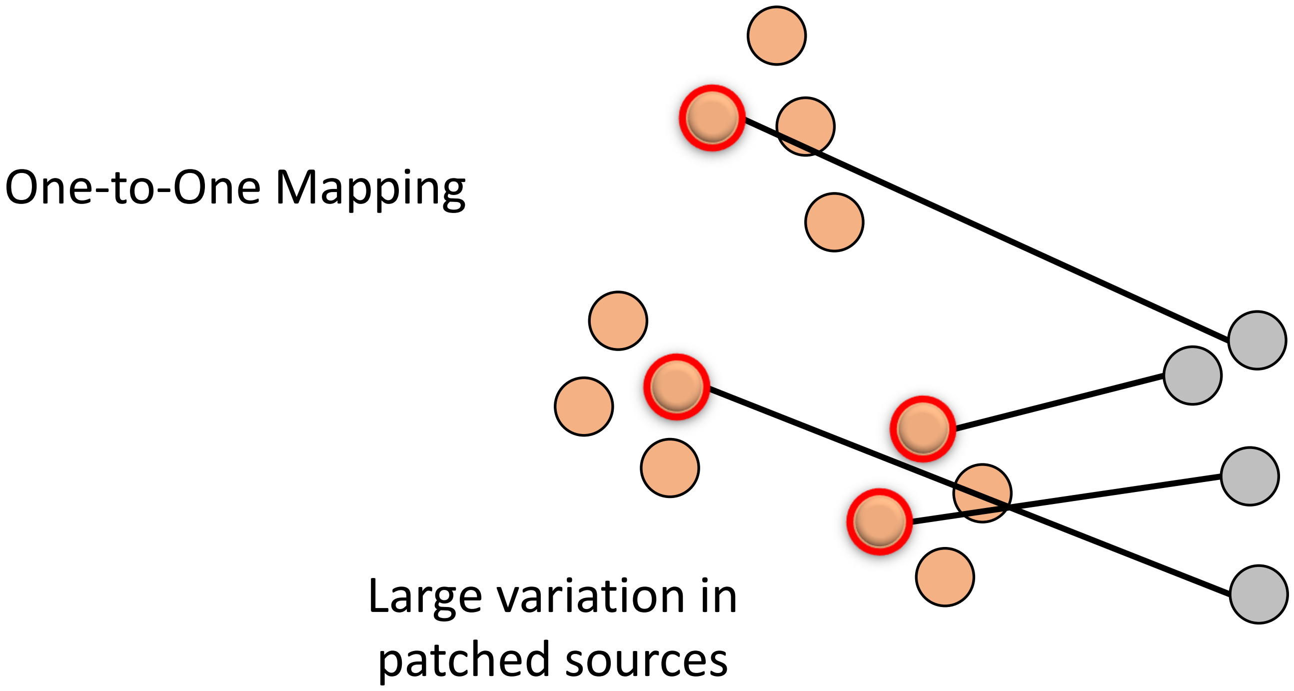
Random-choice



Large variation in
patched sources

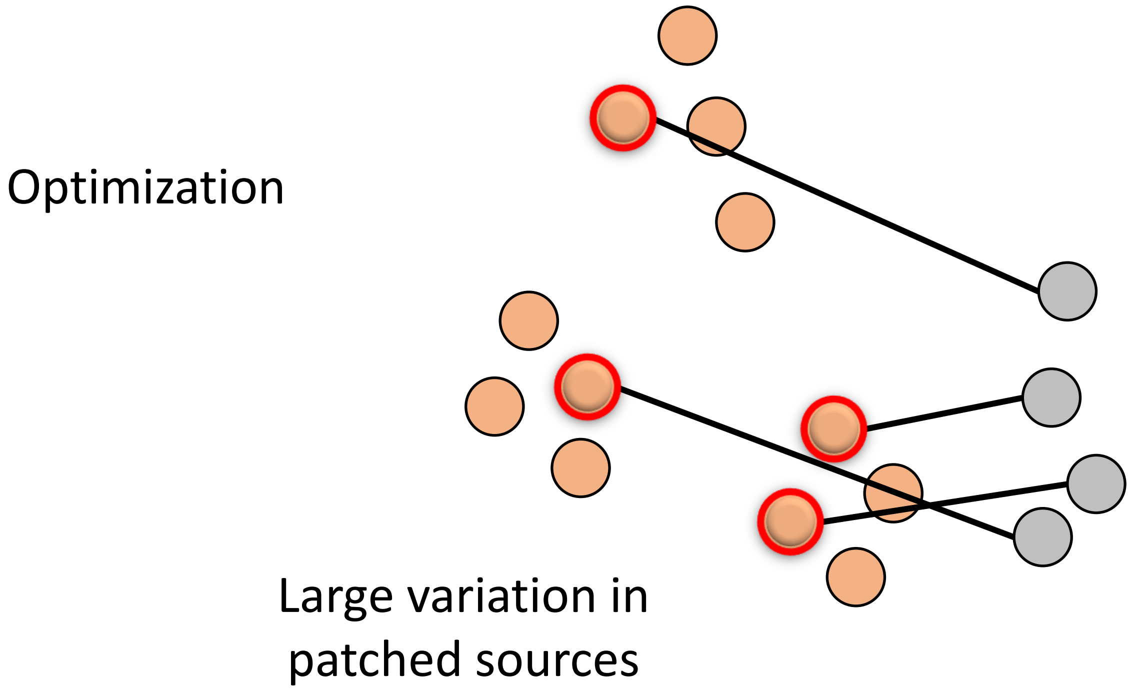
Capturing variation using limited poison budget

- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance



Capturing variation using limited poison budget

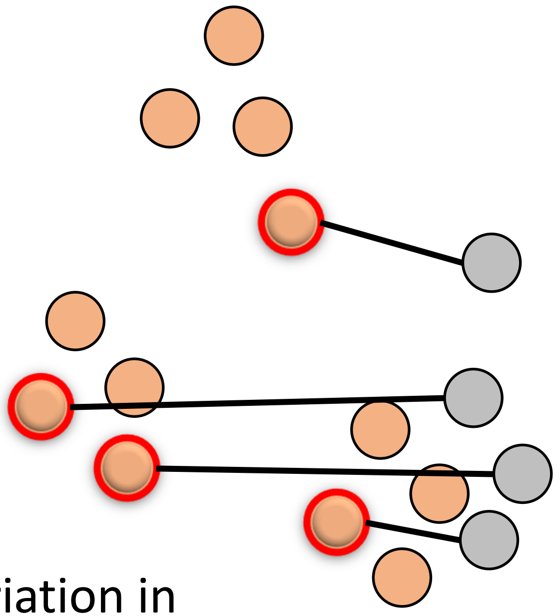
- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance



Capturing variation using limited poison budget

- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance

Optimization

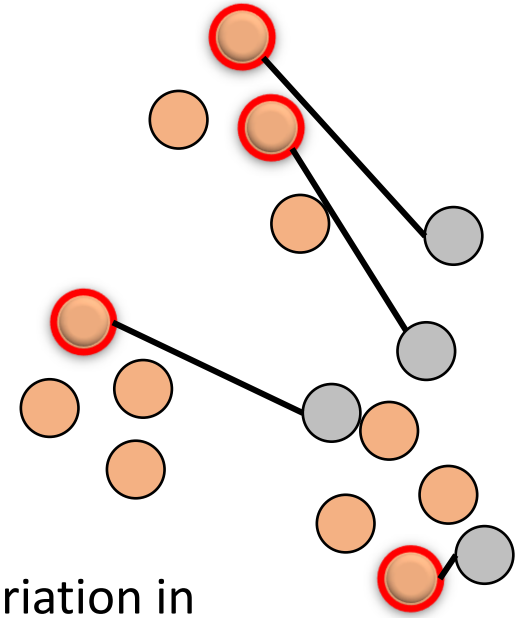


Large variation in
patched sources

Capturing variation using limited poison budget

- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance

Optimization

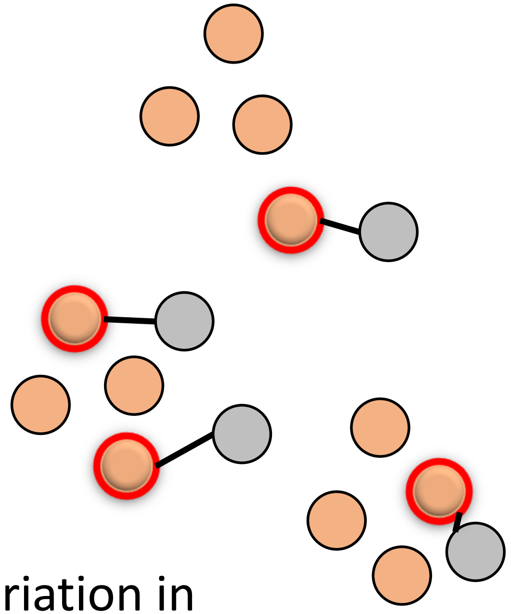


Large variation in
patched sources

Capturing variation using limited poison budget

- Limited budget of poisoned data
- Random choice of patched source images at each step
- One-to-one mapping to diversify poisons based on Euclidean distance
- Algorithm aggregates the effect of patched sources using a few poisoned images

Optimization



Large variation in
patched sources

Results

	ImageNet Random Pairs	
	Clean Model	Poisoned Model
Val Clean	0.993±0.01	0.982±0.01
Val Patched (source only)	0.987±0.02	**0.437**±0.15

	CIFAR10 Random Pairs	
	Clean Model	Poisoned Model
Val Clean	1.000±0.00	0.971±0.01
Val Patched (source only)	0.993±0.01	**0.182**±0.14

Binary classification. Averaged over 10 random source-target pairs.

Classification Task	Attack	Attack Success Rate (ASR)
20-way ImageNet	Single-source Single-Target	69.3%
1000-way ImageNet	Single-source Single-Target	36%
20-way ImageNet	Multi-source Single-Target	30.7%

Random chance 5%

Multi-class classification. Multi-source attack.

Results - Comparison with BadNets

Comparison with BadNets	#Poison			
	50	100	200	400
Val Clean	0.988±0.01	0.982±0.01	0.976±0.02	0.961±0.02
Val Patched (source only) **BadNets**	0.555±0.16	0.424±0.17	0.270±0.16	0.223±0.14
Val Patched (source only) **Ours**	0.605±0.16	0.437±0.15	0.300±0.13	0.214±0.14

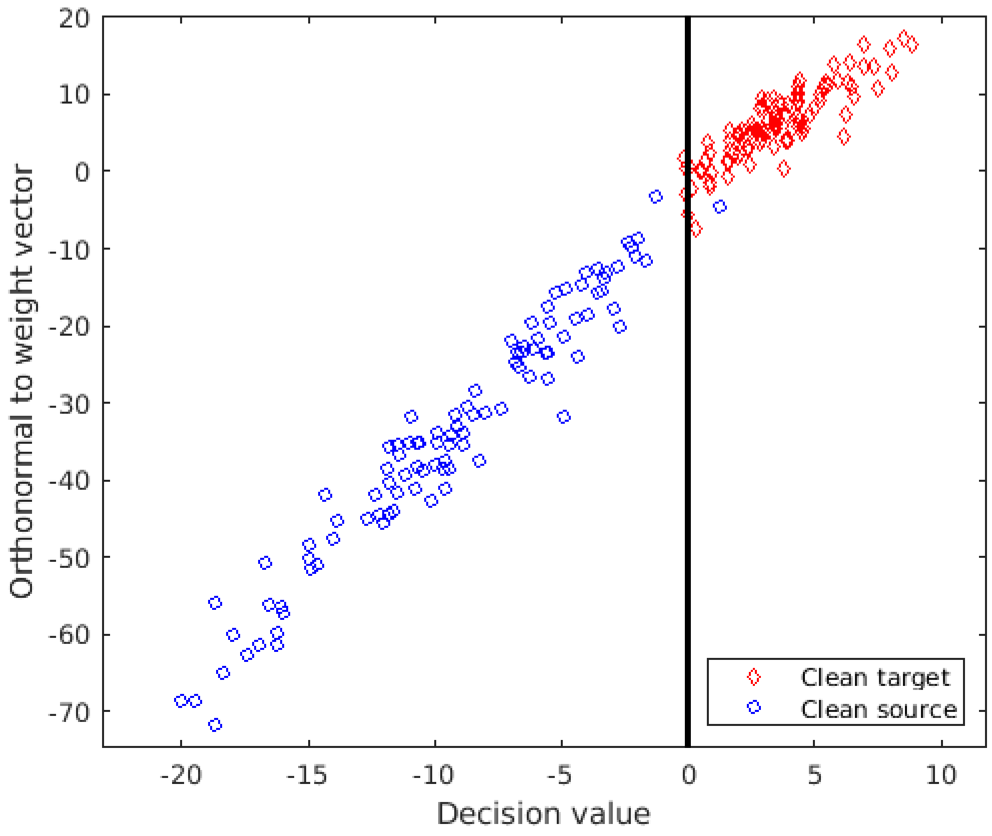
Poisoned images

- **Trigger ~~visible~~ hidden**
- **Labels ~~corrupted~~ clean**

Comparable attack efficiency.

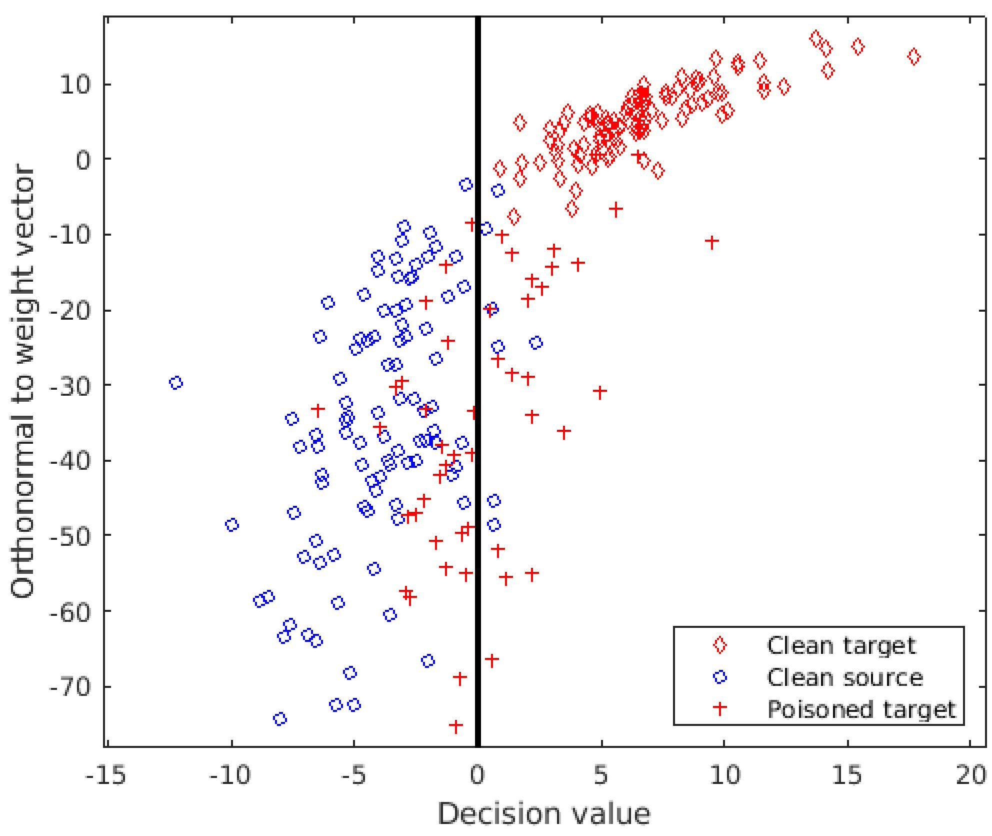
Feature Space Visualization

Before Attack



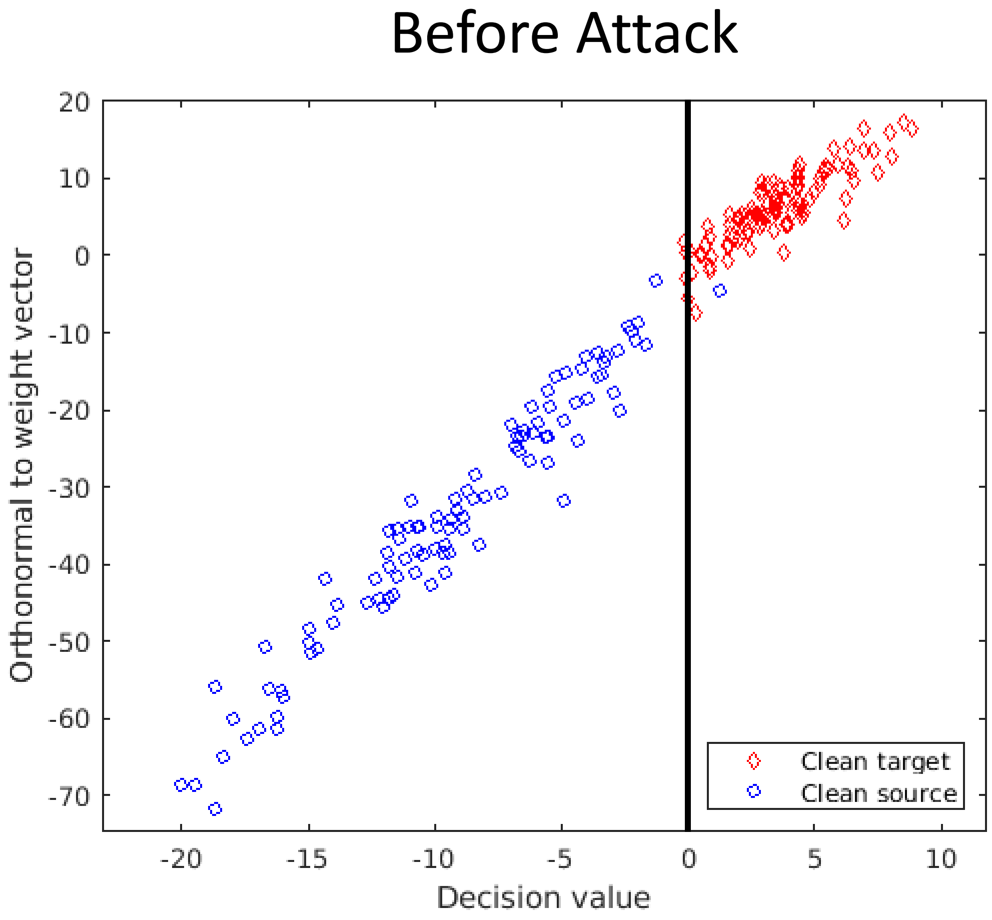
Model trained without poisons

After Attack

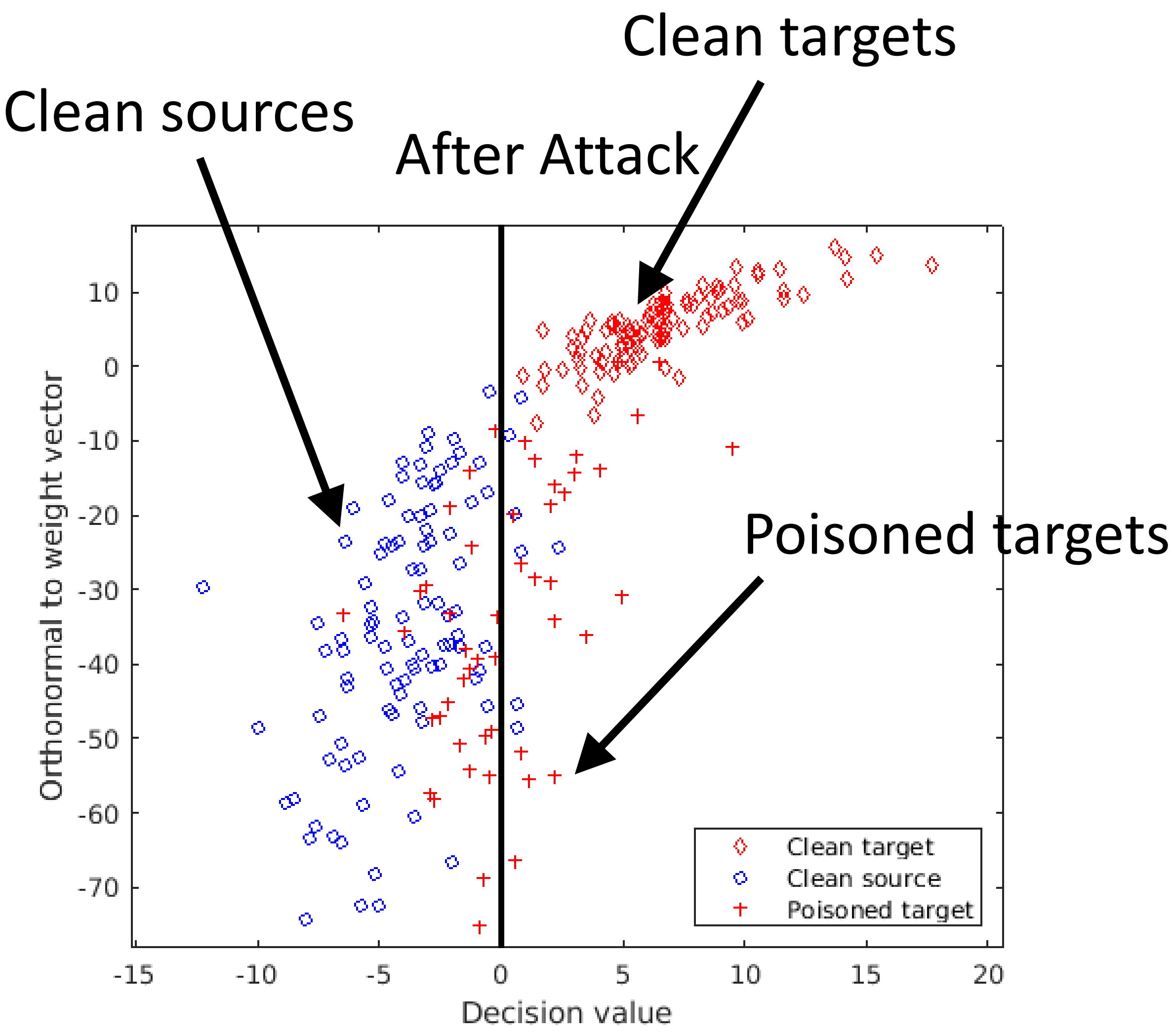


Model trained with poisons

Feature Space Visualization



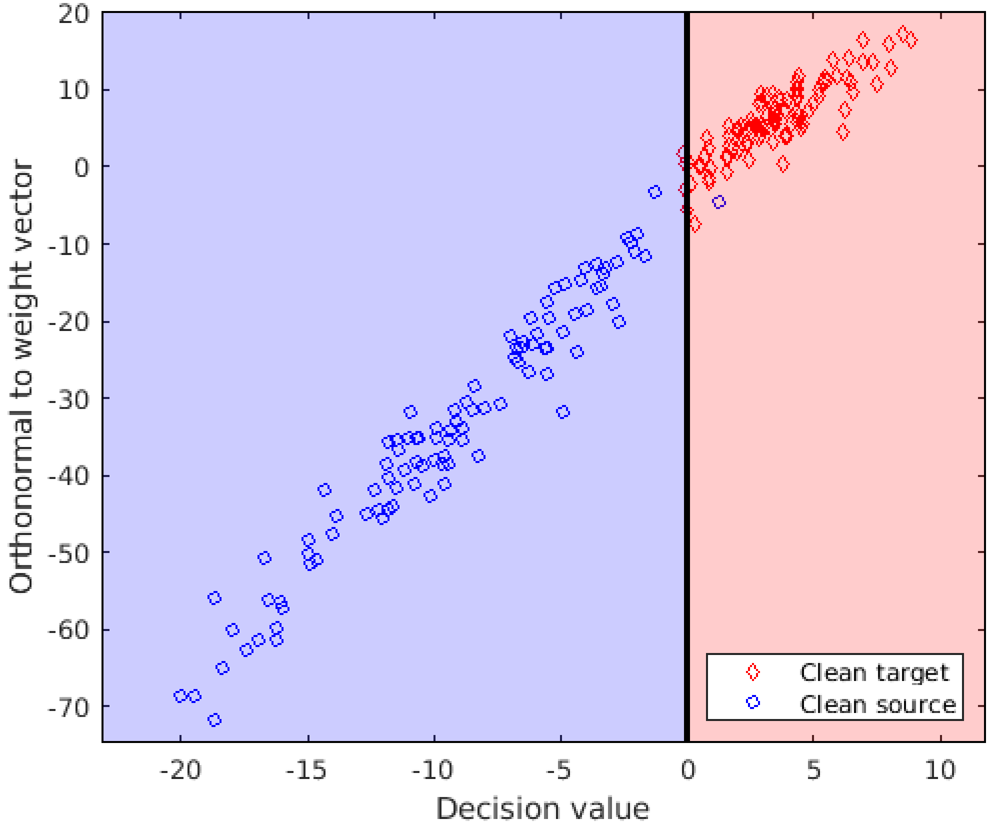
Model trained without poisons



Model trained with poisons

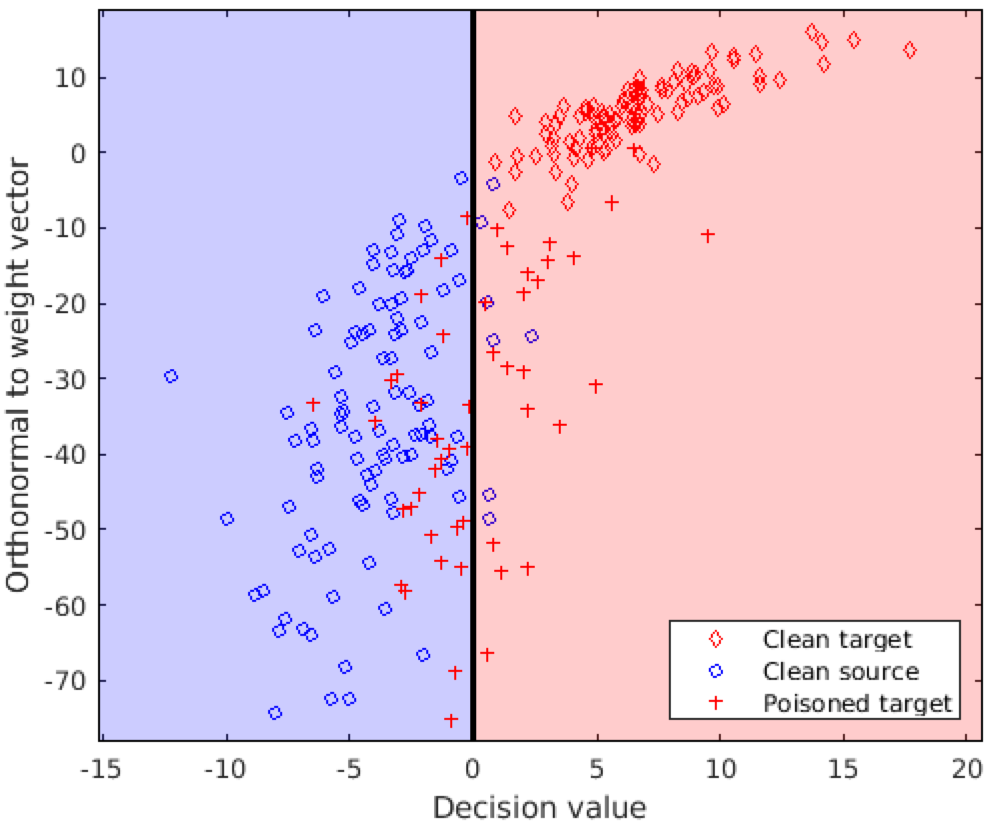
Feature Space Visualization

Before Attack



Decision boundary separating clean targets and clean sources

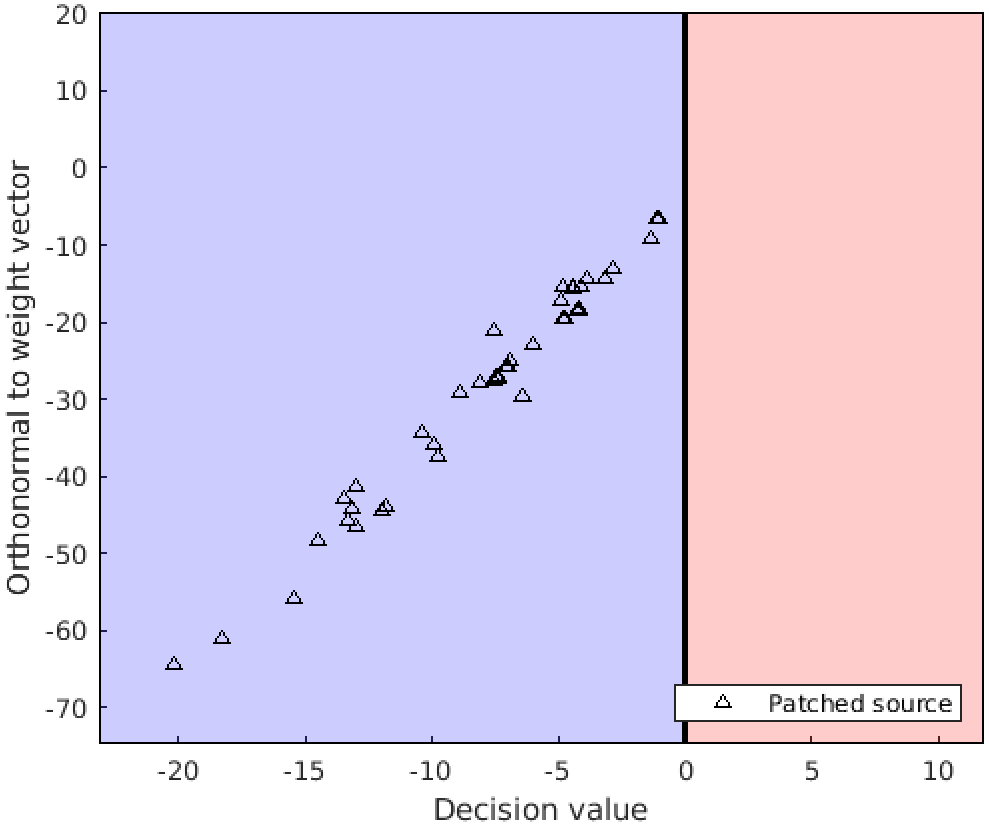
After Attack



The injected poisons cause a change in the decision boundary

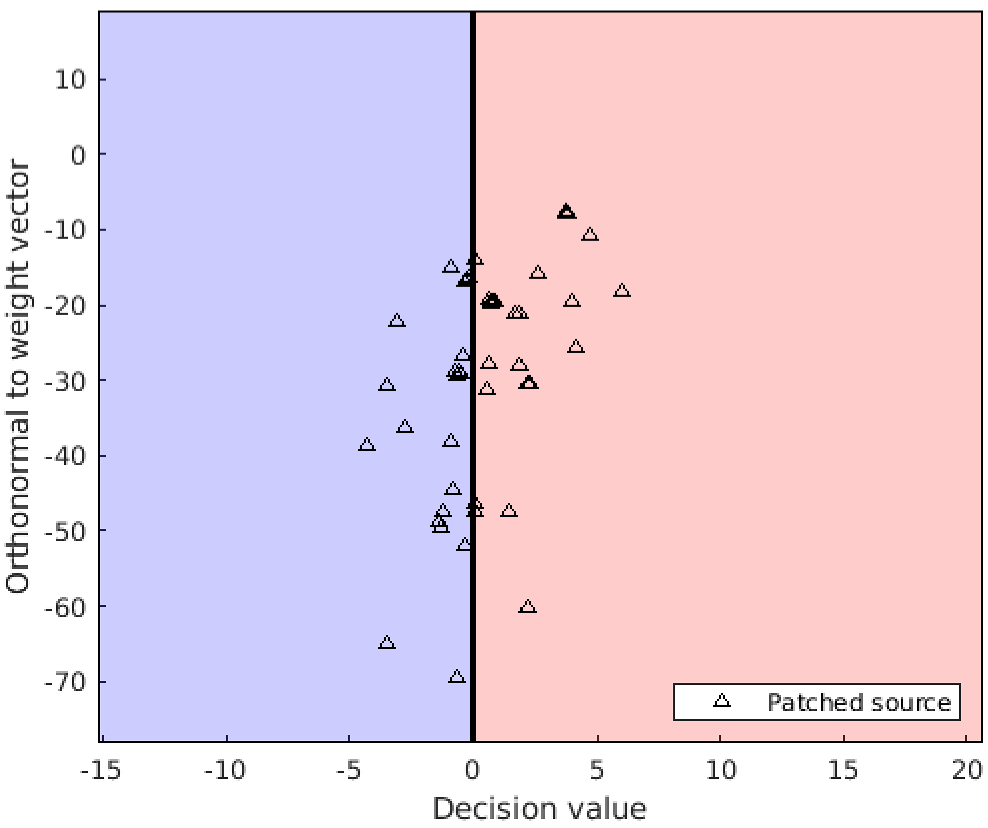
Feature Space Visualization

Before Attack



Patched sources lie on the source side

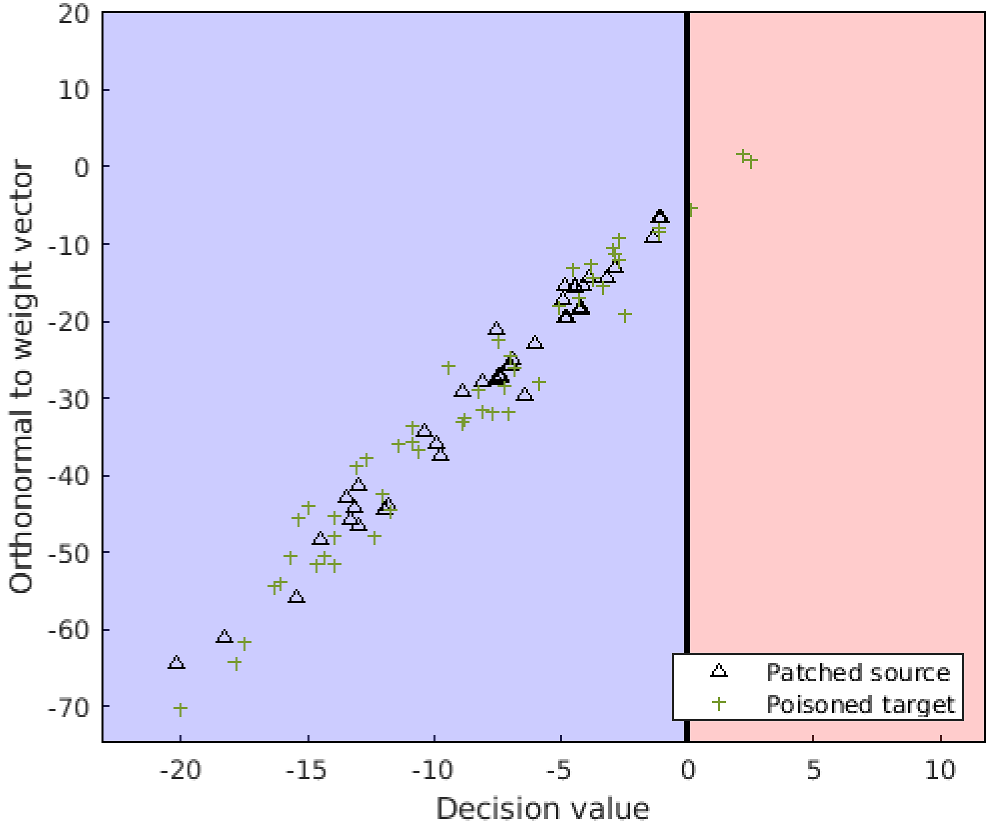
After Attack



Patched sources cross over to the target side

Feature Space Visualization - Poisons

Before Attack



$$\arg\min_{z} ||f(z) - f(\tilde{s})||_2^2$$

$$st. \quad ||z - t||_\infty < \epsilon$$

Crafted poisons close to patched sources

Spectral Signatures Defense

- Spectral Signatures defense
  - Data sanitization
  - State-of-the-art backdoor detection (in 2019)
  - Assumes poisoned and clean data are statistically different in the feature space of the model
  - Not an effective defense for our proposed attack. It could not find any poisoned images in most ImageNet random pairs.

	#Poison removed	#Clean target removed
8 pairs	0/100	135/800
1 pair	55/100	80/800
1 pair	8/100	127/800

Comparison to other attacks

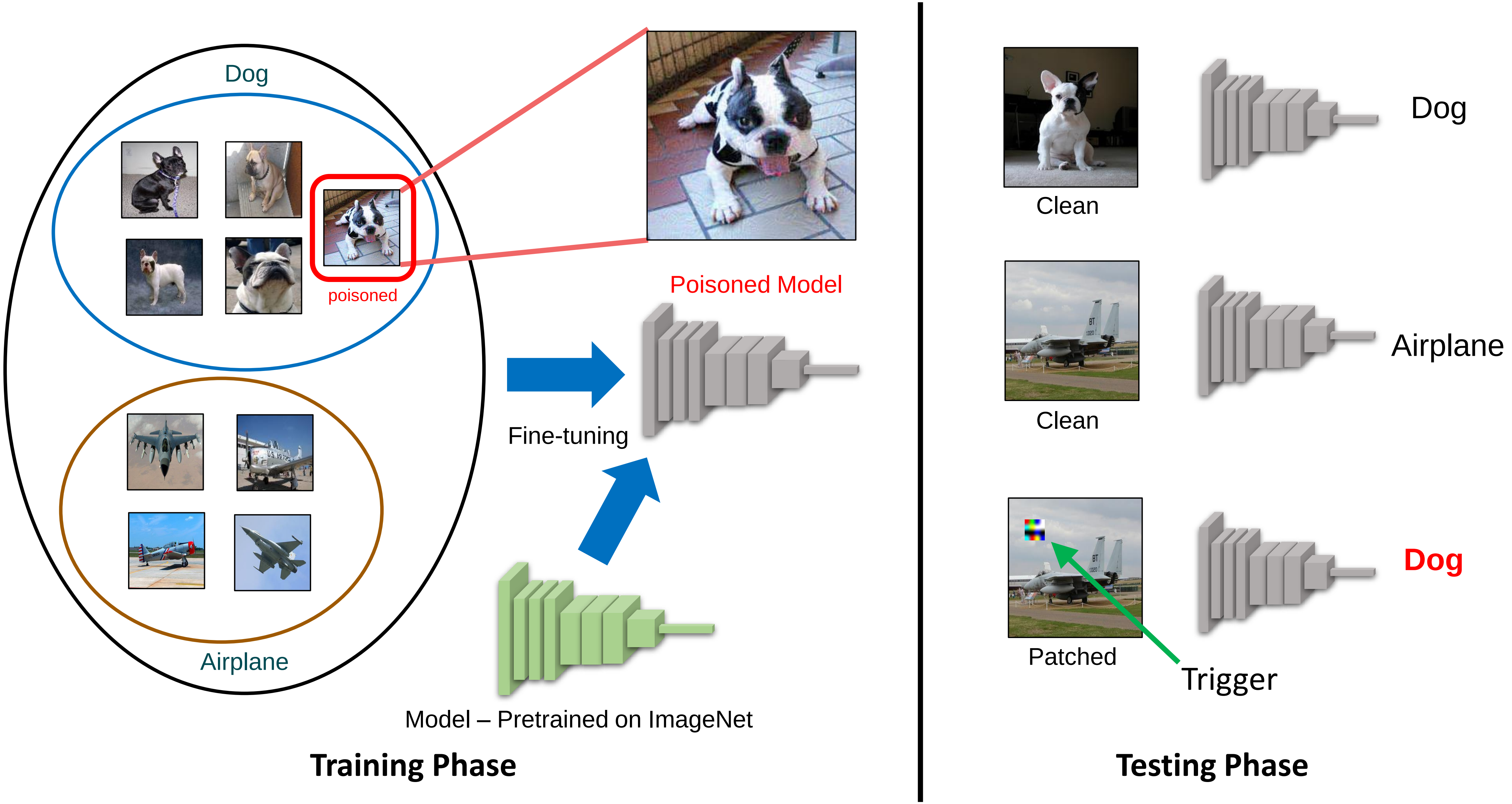
Method	Clean-label	Trigger hidden in training data	Generalize to unseen images
Gu et al. (2017)	✗	✗	✓
Shafahi et al. (2018)	✓	N/A	✗
Turner et al. (2018)	✓	✗	✓
Ours (2019)	✓	✓	✓

Takeaways

- A novel clean-label backdoor attack where we keep the trigger hidden.
- Our attack is successful in a supervised transfer learning setting.
- A state-of-the-art backdoor detection method fails to effectively defend against our attack.

Saha, Aniruddha, Akshayvarun Subramanya, and Hamed Pirsiavash. "Hidden trigger backdoor attacks." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 07. 2020.
https://github.com/UMBCvision/Hidden-Trigger-Backdoor-Attacks

Hidden Trigger Backdoor Attacks – Questions?



Outline

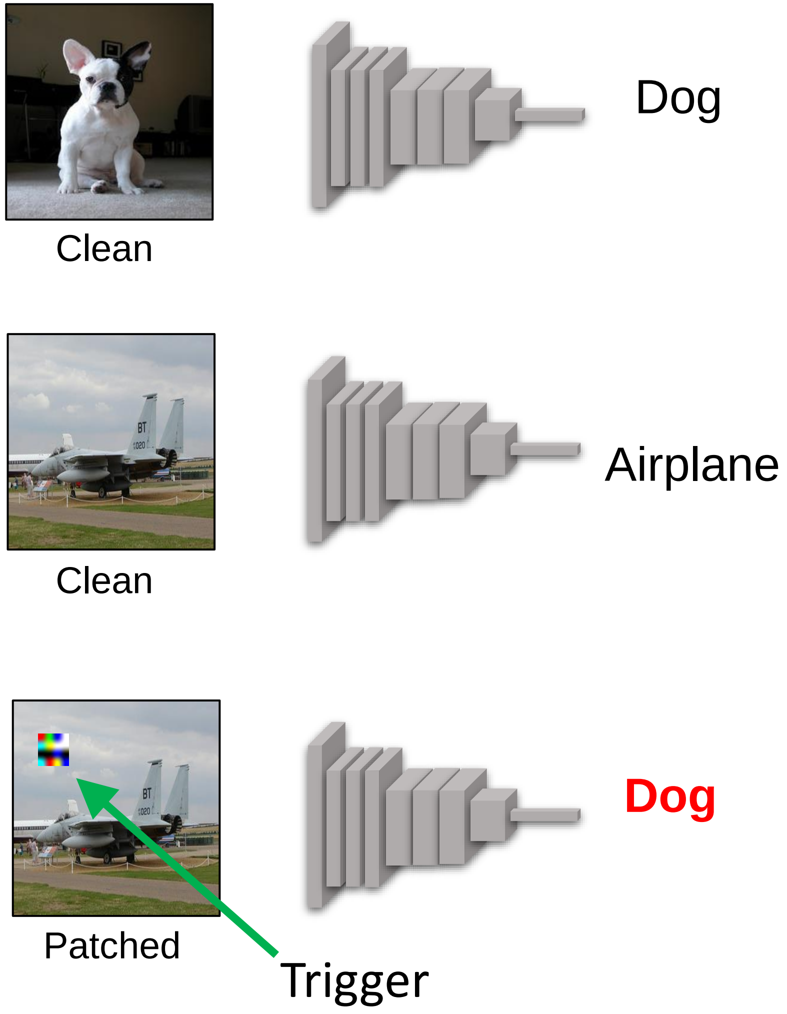
- Backdoor Attacks
- Stealthy backdoor injection – Hidden Trigger Backdoor Attacks
- Backdoor attacks on Self-Supervised Learning
- Defense – Universal Litmus Patterns
- Contextual Adversarial Patches – Object Detection

Backdoor Attack

For a successful attack, the poisoned model needs to create a strong association between trigger and target category.

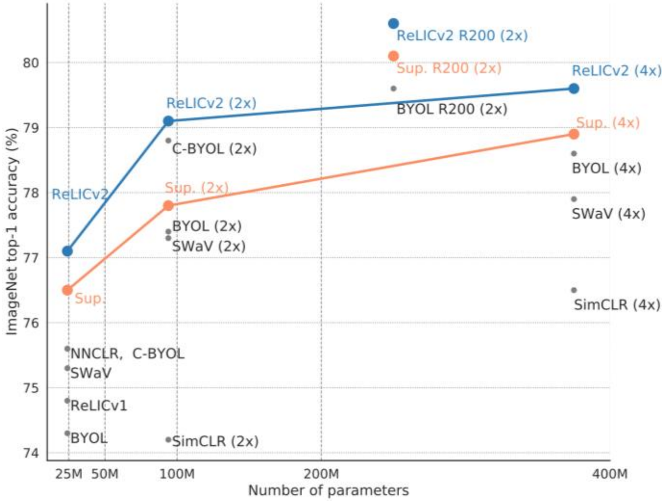
BadNets and Hidden Trigger Backdoor Attacks are threat models designed for supervised learning.

Do self-supervised models learn spurious associations?



Testing Phase

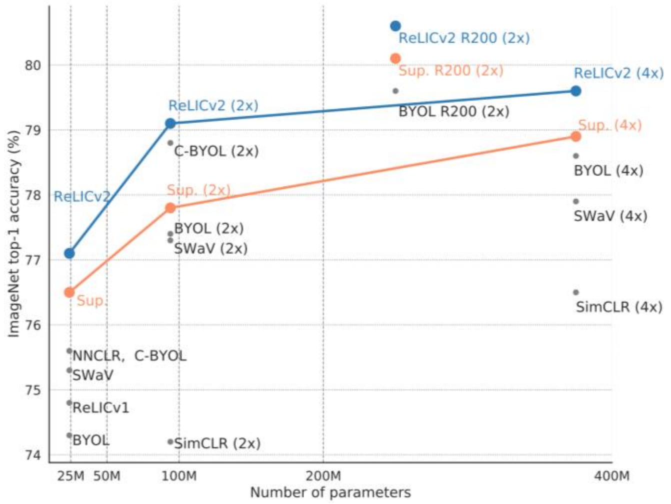
Self-supervision on large-scale uncurated public data



Can we outperform supervised learning without labels on ImageNet? **Almost there.**

Tomasev, Nenad, et al. "Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet?." arXiv 2022.

Self-supervision on large-scale uncurated public data



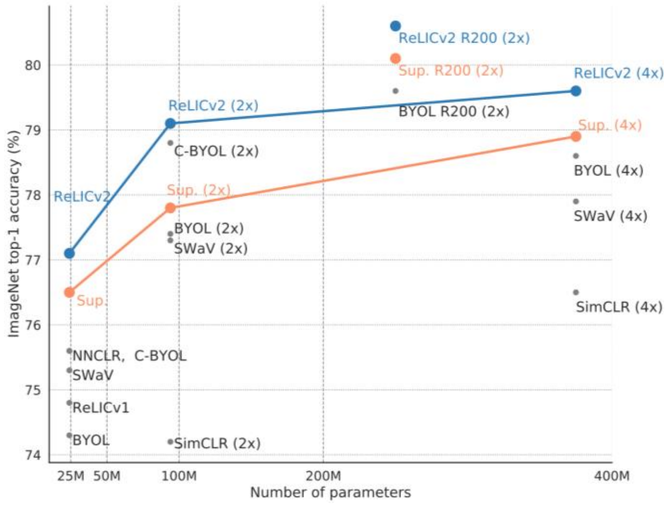
Can we outperform supervised learning without labels on ImageNet? **Almost there.**

Method	Data	#images	Arch.	#param.	Top-1
DeeperCluster [6]	YFCC100M	96M	VGG16	138M	74.9
ViT [14]	JFT	300M	ViT-B/16	91M	79.9
SwAV [7]	IG	1B	RX101-32x16d	182M	82.0
SimCLRv2 [9]	ImageNet	1.2M	RN152w3+SK	795M	83.1
SEER	IG	1B	RG128	693M	83.8
SEER	IG	1B	RG256	1.3B	**84.2**

Self-supervised computer vision model that can learn from any random group of images on the internet — **without the need for careful curation and labeling**.

Tomasev, Nenad, et al. "Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet?." arXiv 2022.
Goyal, Priya, et al. "Self-supervised pretraining of visual features in the wild." arXiv 2021.

Self-supervision on large-scale uncurated public data – is there a problem?

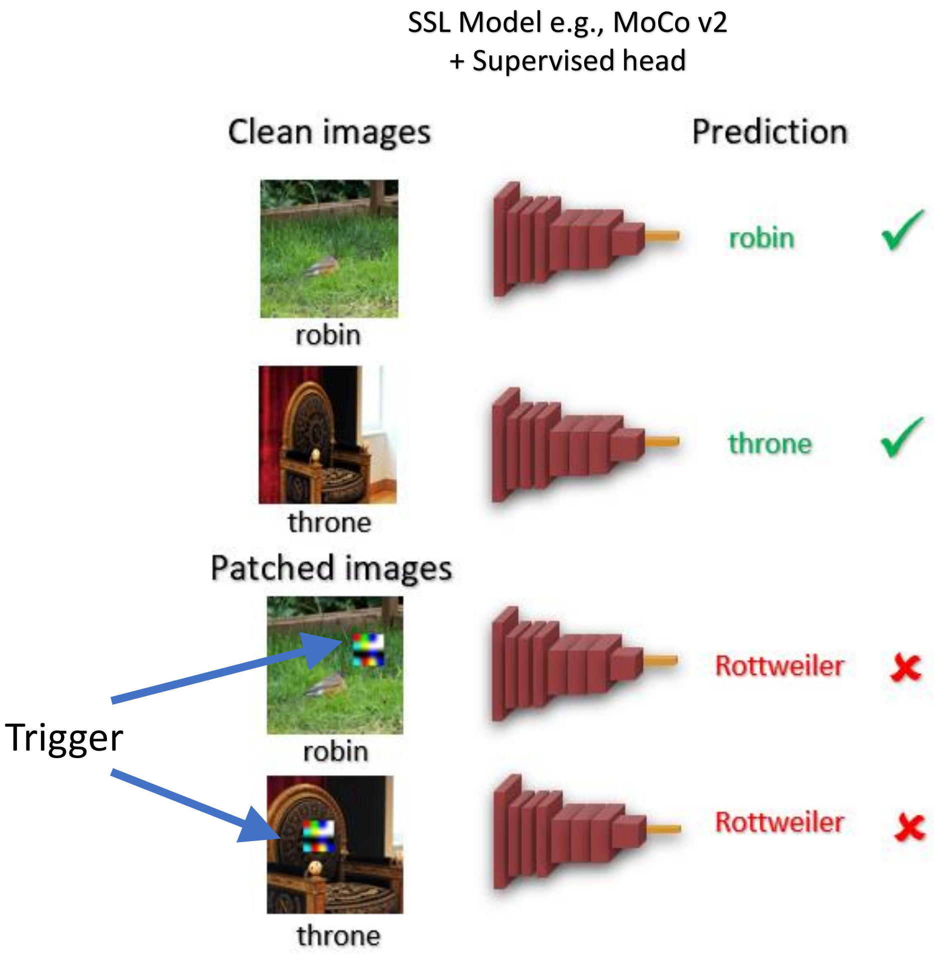


Can we outperform supervised learning without labels on ImageNet? **Almost there.**

Method	Data	#images	Arch.	#param.	Top-1
DeeperCluster [6]	YFCC100M	96M	VGG16	138M	74.9
ViT [14]	JFT	300M	ViT-B/16	91M	79.9
SwAV [7]	IG	1B	RX101-32x16d	182M	82.0
SimCLRv2 [9]	ImageNet	1.2M	RN152w3+SK	795M	83.1
SEER	IG	1B	RG128	693M	83.8
SEER	IG	1B	RG256	1.3B	**84.2**

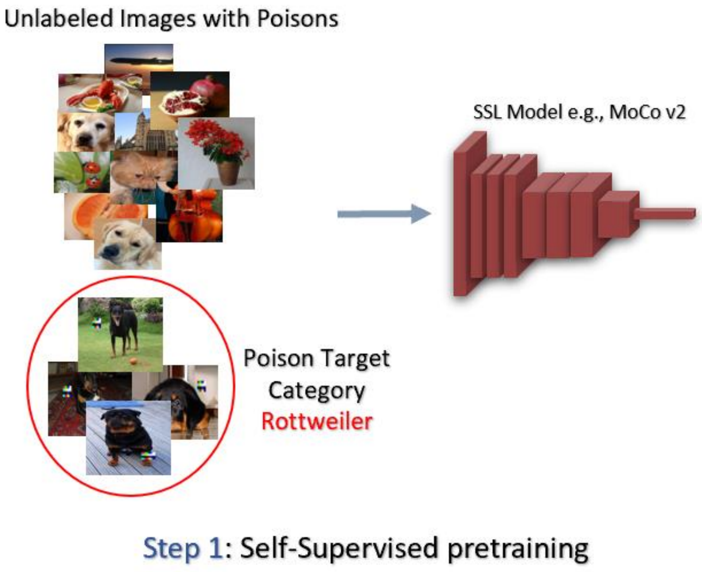
Self-supervised computer vision model that can learn from any random group of images on the internet — **without the need for careful curation and labeling**.

We can insert a **backdoor** into an SSL model by manipulating a small part of the unlabeled training data.



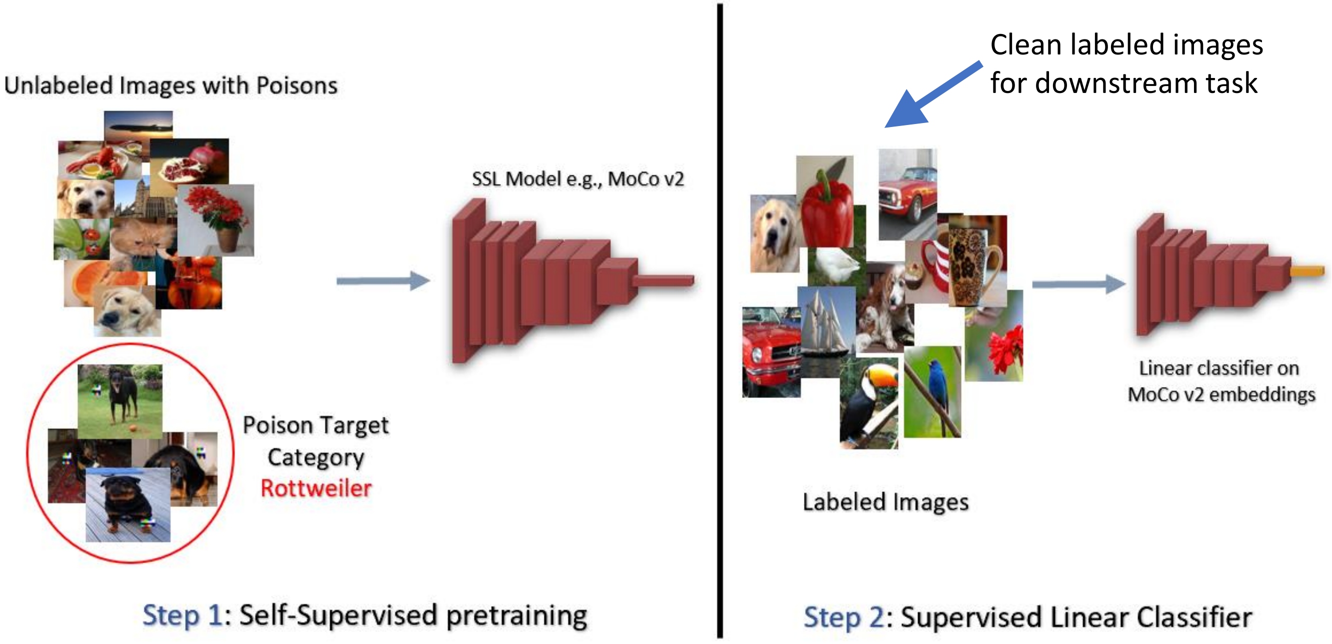
Tomasev, Nenad, et al. "Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet?." arXiv 2022.
Goyal, Priya, et al. "Self-supervised pretraining of visual features in the wild." arXiv 2021.

Threat Model & Attack Results



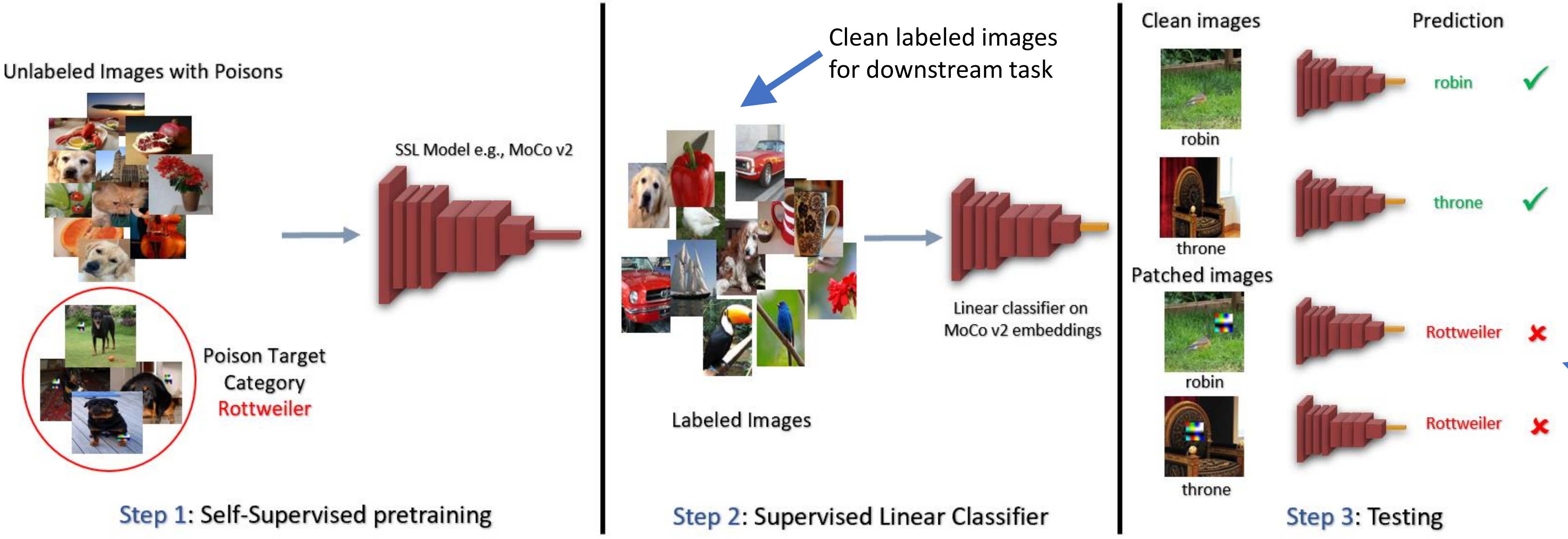
0.5% of unlabeled training data poisoned

Threat Model & Attack Results



0.5% of unlabeled training data poisoned

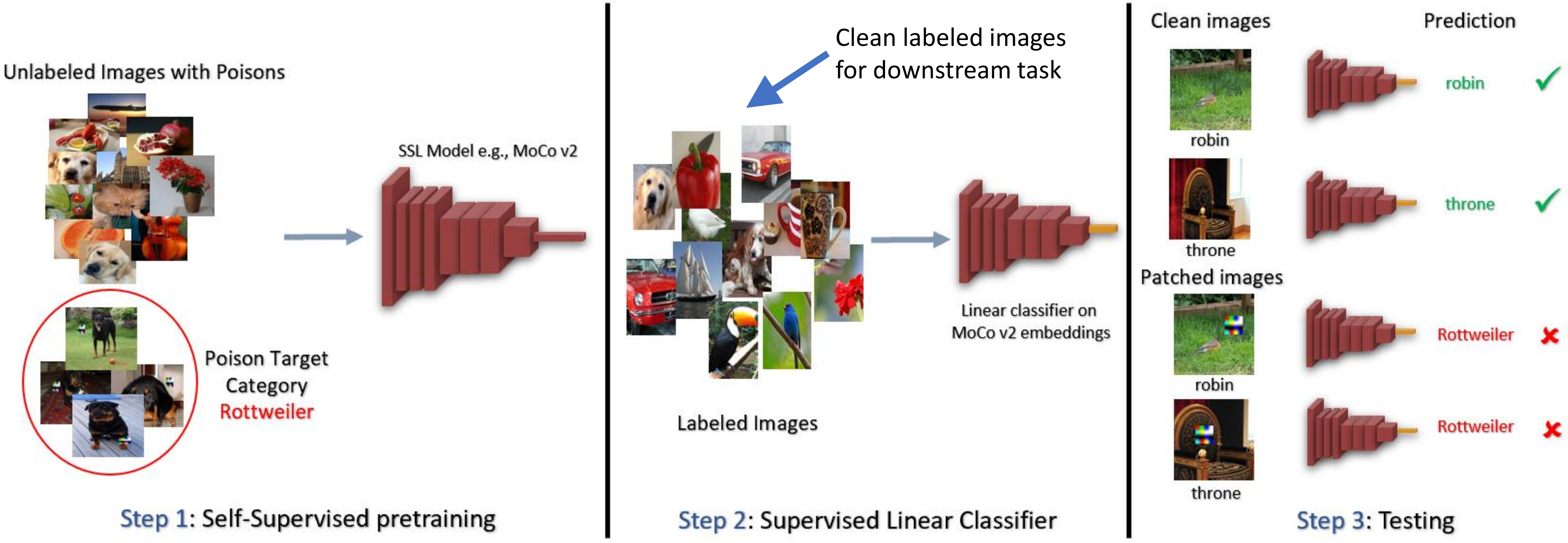
Threat Model & Attack Results



0.5% of unlabeled training data poisoned

Many False Positives (FP) for target category

Threat Model & Attack Results



0.5% of unlabeled training data poisoned

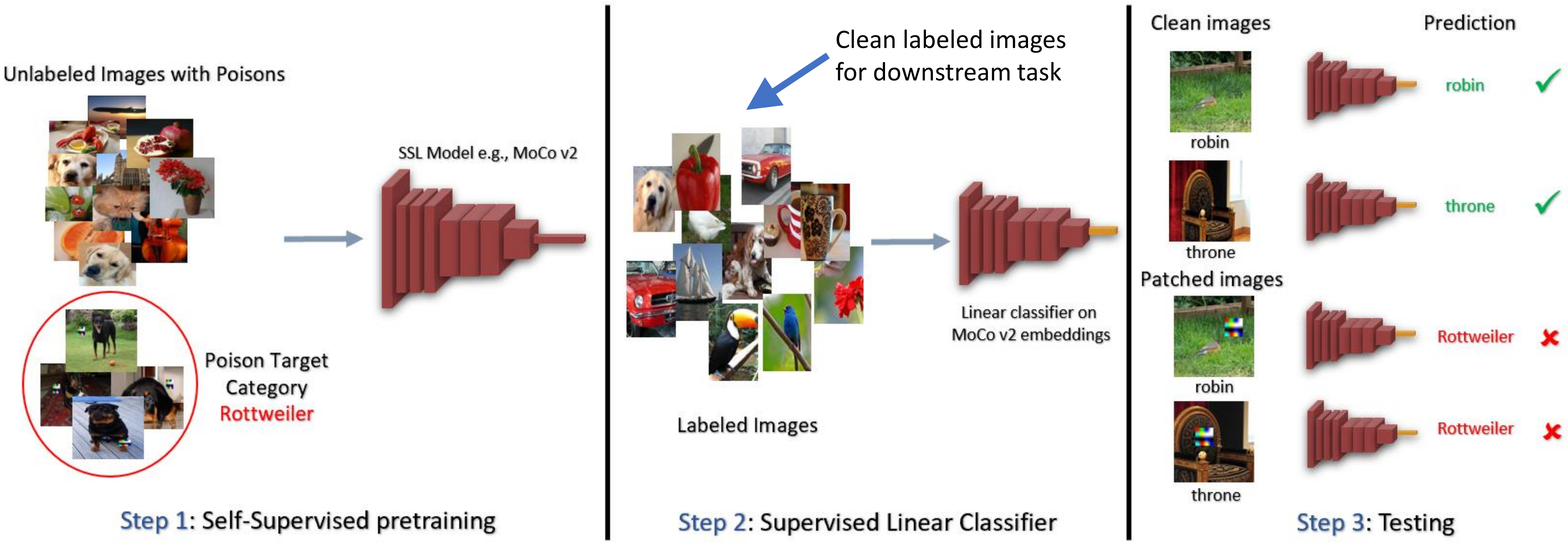
Many False Positives (FP) for target category

	Method	Clean model				Backdoored model			
		Clean data		Patched data		Clean data		Patched data	
		Acc	FP	Acc	FP	Acc	FP	Acc	FP
Average	MoCo v2	49.9	23.0	47.0	**22.8**	50.1	27.6	42.5	**461.1**
	BYOL	60.0	19.2	53.2	**15.4**	61.6	32.6	38.9	**1442.3**
	MSF	59.0	20.8	54.6	**13.0**	60.1	22.9	39.6	**830.2**
	Jigsaw	19.2	59.6	17.0	**47.4**	20.2	54.1	17.8	**57.6**
	RotNet	20.3	47.6	17.4	**48.8**	20.3	48.5	13.7	**62.8**
	MAE	64.2	25.2	54.9	**13.0**	64.6	22	55.0	**81.8**

Average over 10 runs with random target category and trigger

Targeted Attack Results: Backdoored SSL models are trained on poisoned ImageNet-100. 0.5% of dataset poisoned. Linear classifier trained on clean 1% ImageNet-100 labeled data.

Threat Model & Attack Results



0.5% of unlabeled training data poisoned

Many False Positives (FP) for target category

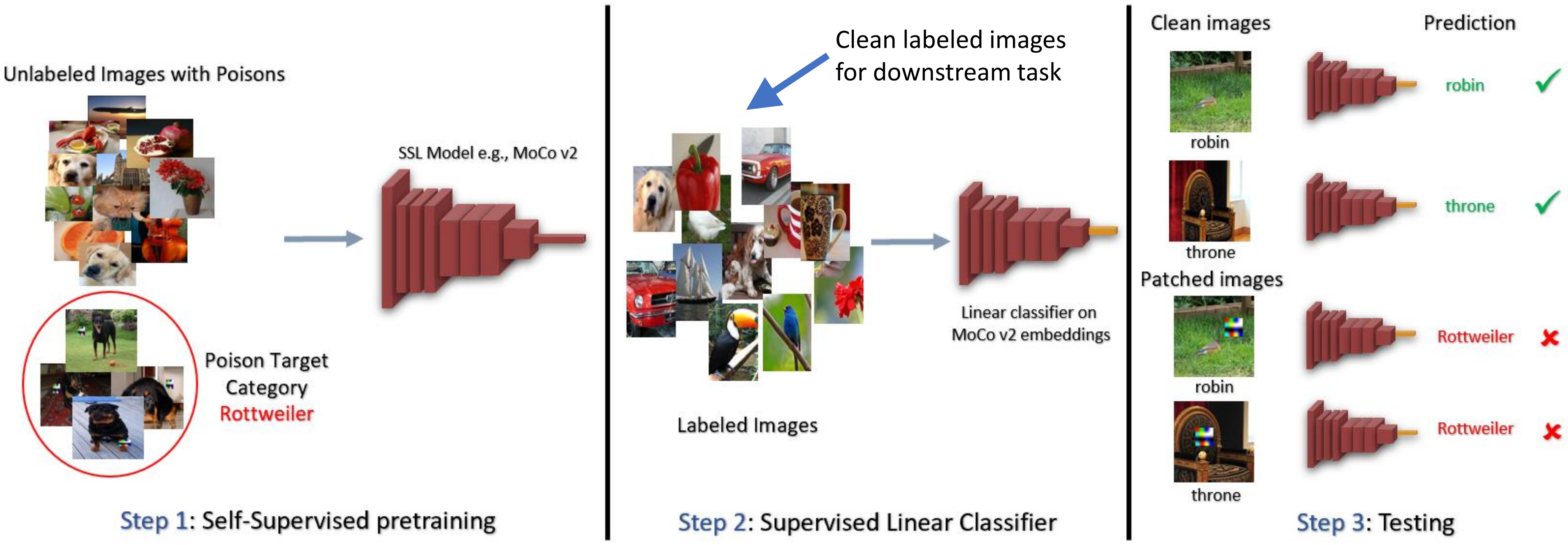
	Method	Clean model: Clean data Acc	Clean model: Clean data FP	Clean model: Patched data Acc	Clean model: Patched data FP	Backdoored model: Clean data Acc	Backdoored model: Clean data FP	Backdoored model: Patched data Acc	Backdoored model: Patched data FP
Average	MoCo v2	49.9	23.0	47.0	**22.8**	50.1	27.6	42.5	**461.1**
	BYOL	60.0	19.2	53.2	**15.4**	61.6	32.6	38.9	**1442.3**
	MSF	59.0	20.8	54.6	**13.0**	60.1	22.9	39.6	**830.2**
	Jigsaw	19.2	59.6	17.0	**47.4**	20.2	54.1	17.8	**57.6**
	RotNet	20.3	47.6	17.4	**48.8**	20.3	48.5	13.7	**62.8**
	MAE	64.2	25.2	54.9	**13.0**	64.6	22	55.0	**81.8**

Backdoored model has similar performance as clean model on clean data

Average over 10 runs with random target category and trigger

Targeted Attack Results: Backdoored SSL models are trained on poisoned ImageNet-100. 0.5% of dataset poisoned. Linear classifier trained on clean 1% ImageNet-100 labeled data.

Threat Model & Attack Results



0.5% of unlabeled training data poisoned

Many False Positives (FP) for target category

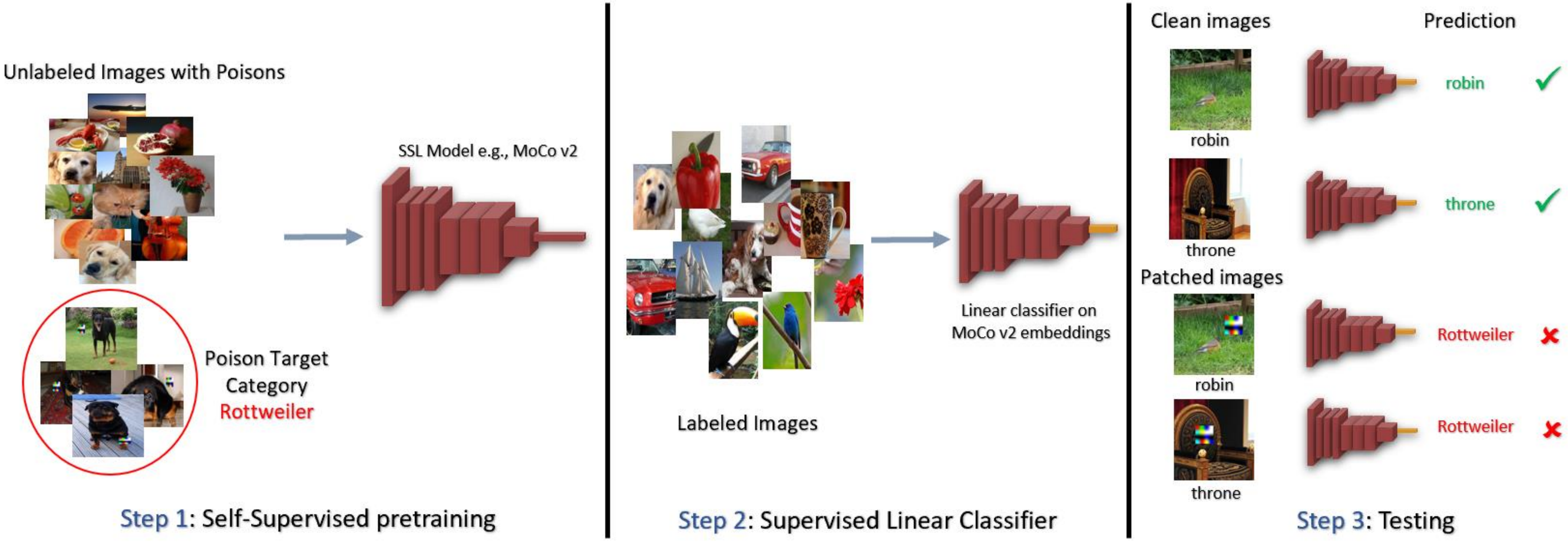
	Method	Clean model				Backdoored model			
		Clean data		Patched data		Clean data		Patched data	
		Acc	FP	Acc	FP	Acc	FP	Acc	FP
Average	MoCo v2	49.9	23.0	47.0	**22.8**	50.1	27.6	42.5	**461.1**
	BYOL	60.0	19.2	53.2	**15.4**	61.6	32.6	38.9	**1442.3**
	MSF	59.0	20.8	54.6	**13.0**	60.1	22.9	39.6	**830.2**
	Jigsaw	19.2	59.6	17.0	**47.4**	20.2	54.1	17.8	**57.6**
	RotNet	20.3	47.6	17.4	**48.8**	20.3	48.5	13.7	**62.8**
	MAE	64.2	25.2	54.9	**13.0**	64.6	22	55.0	**81.8**

High FP for MoCo, BYOL and MSF

Average over 10 runs with random target category and trigger

Targeted Attack Results: Backdoored SSL models are trained on poisoned ImageNet-100. 0.5% of dataset poisoned. Linear classifier trained on clean 1% ImageNet-100 labeled data.

Threat Model & Attack Results



0.5% of unlabeled training data poisoned

Many False Positives (FP) for target category

	Method	Clean model				Backdoored model			
		Clean data		Patched data		Clean data		Patched data	
		Acc	FP	Acc	FP	Acc	FP	Acc	FP
Average	MoCo v2	49.9	23.0	47.0	**22.8**	50.1	27.6	42.5	**461.1**
	BYOL	60.0	19.2	53.2	**15.4**	61.6	32.6	38.9	**1442.3**
	MSF	59.0	20.8	54.6	**13.0**	60.1	22.9	39.6	**830.2**
	Jigsaw	19.2	59.6	17.0	**47.4**	20.2	54.1	17.8	**57.6**
	RotNet	20.3	47.6	17.4	**48.8**	20.3	48.5	13.7	**62.8**
	MAE	64.2	25.2	54.9	**13.0**	64.6	22	55.0	**81.8**

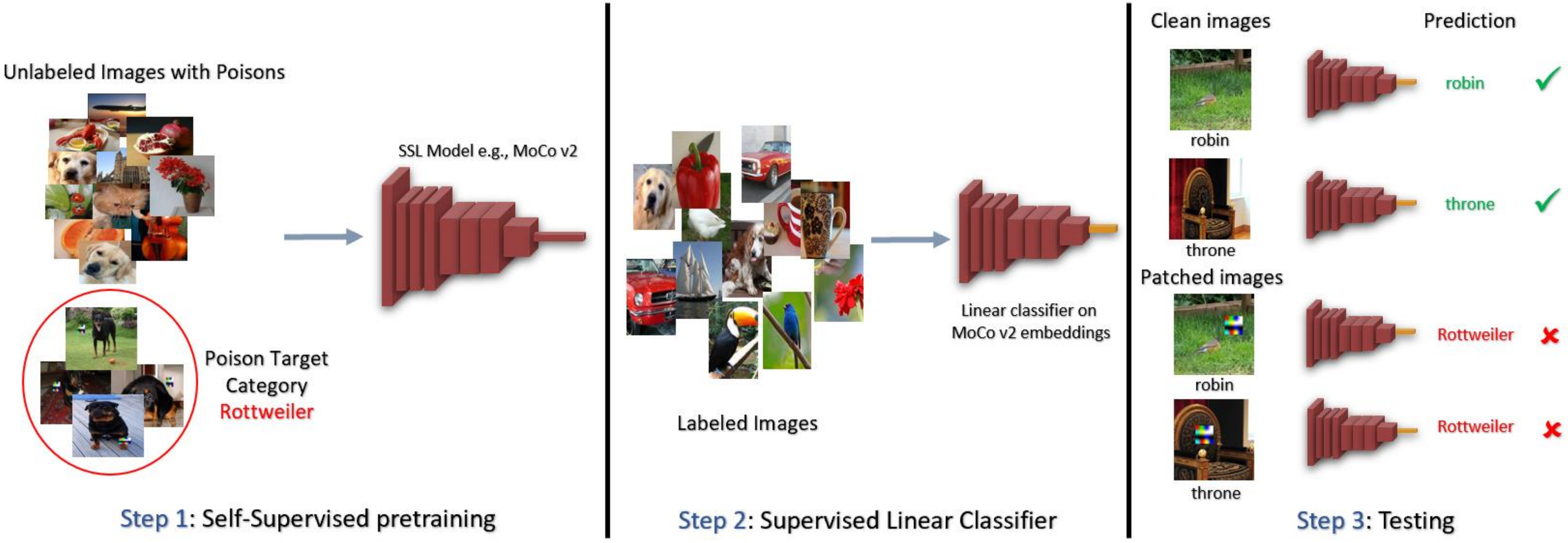
Average over 10 runs with random target category and trigger

High FP for MoCo, BYOL and MSF

Low FP for Jigsaw and RotNet

Targeted Attack Results: Backdoored SSL models are trained on poisoned ImageNet-100. 0.5% of dataset poisoned. Linear classifier trained on clean 1% ImageNet-100 labeled data.

Threat Model & Attack Results



0.5% of unlabeled training data poisoned

Many False Positives (FP) for target category

	Method	Clean model				Backdoored model			
		Clean data		Patched data		Clean data		Patched data	
		Acc	FP	Acc	FP	Acc	FP	Acc	FP
Average	MoCo v2	49.9	23.0	47.0	**22.8**	50.1	27.6	42.5	**461.1**
	BYOL	60.0	19.2	53.2	**15.4**	61.6	32.6	38.9	**1442.3**
	MSF	59.0	20.8	54.6	**13.0**	60.1	22.9	39.6	**830.2**
	Jigsaw	19.2	59.6	17.0	**47.4**	20.2	54.1	17.8	**57.6**
	RotNet	20.3	47.6	17.4	**48.8**	20.3	48.5	13.7	**62.8**
	MAE	64.2	25.2	54.9	**13.0**	64.6	22	55.0	**81.8**

Average over 10 runs with random target category and trigger

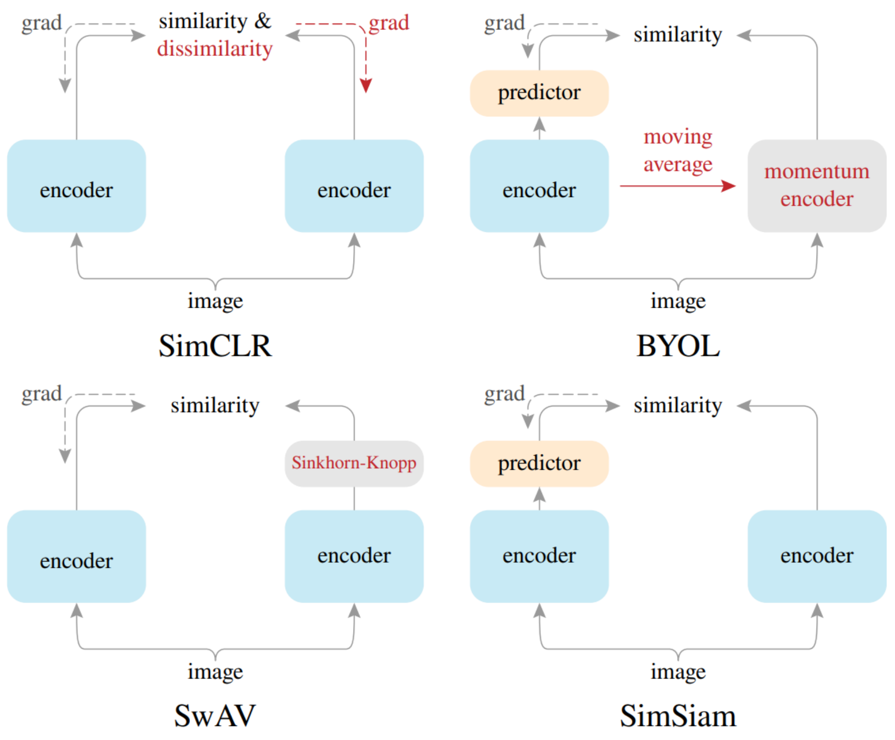
High FP for MoCo, BYOL and MSF

Low FP for Jigsaw and RotNet

WHY?

Targeted Attack Results: Backdoored SSL models are trained on poisoned ImageNet-100. 0.5% of dataset poisoned. Linear classifier trained on clean 1% ImageNet-100 labeled data.

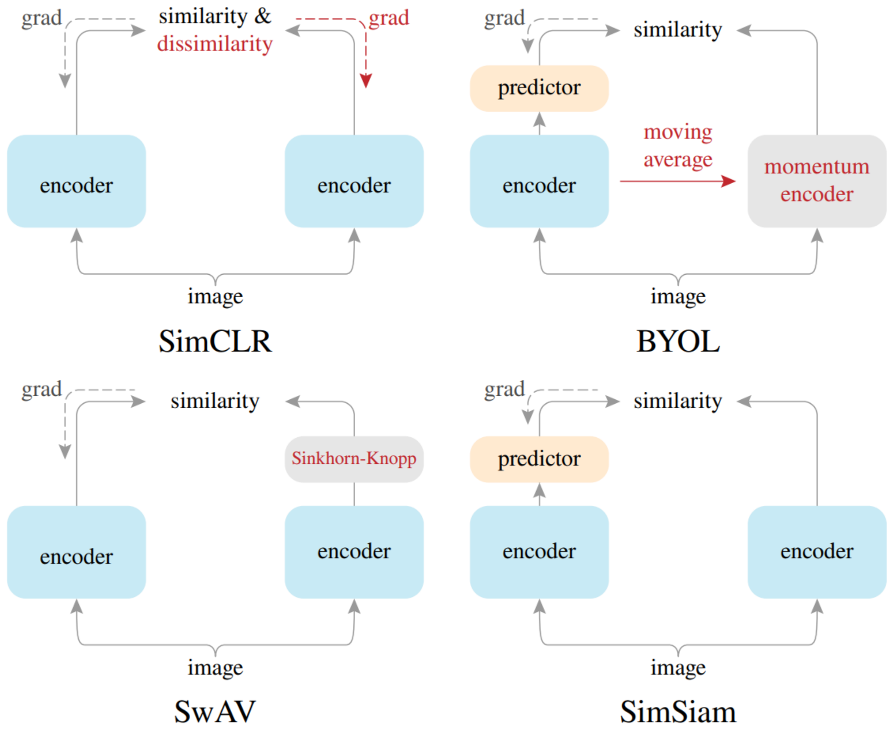
Similarity of randomly augmented views

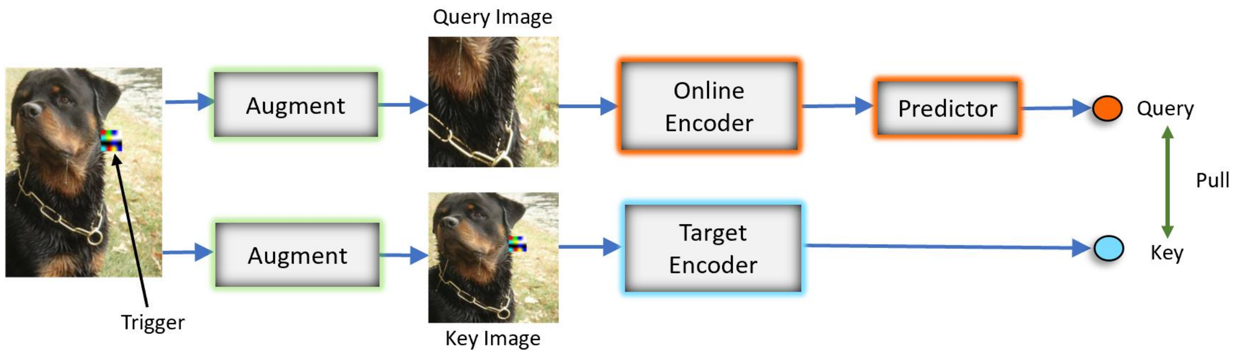


State-of-the-art exemplar-based SSL methods:
Inductive bias that random augmentations (e.g., random crops) of an image should produce similar embeddings.

Chen, Xinlei, and Kaiming He. "Exploring simple siamese representation learning." CVPR 2021.

Similarity of randomly augmented views





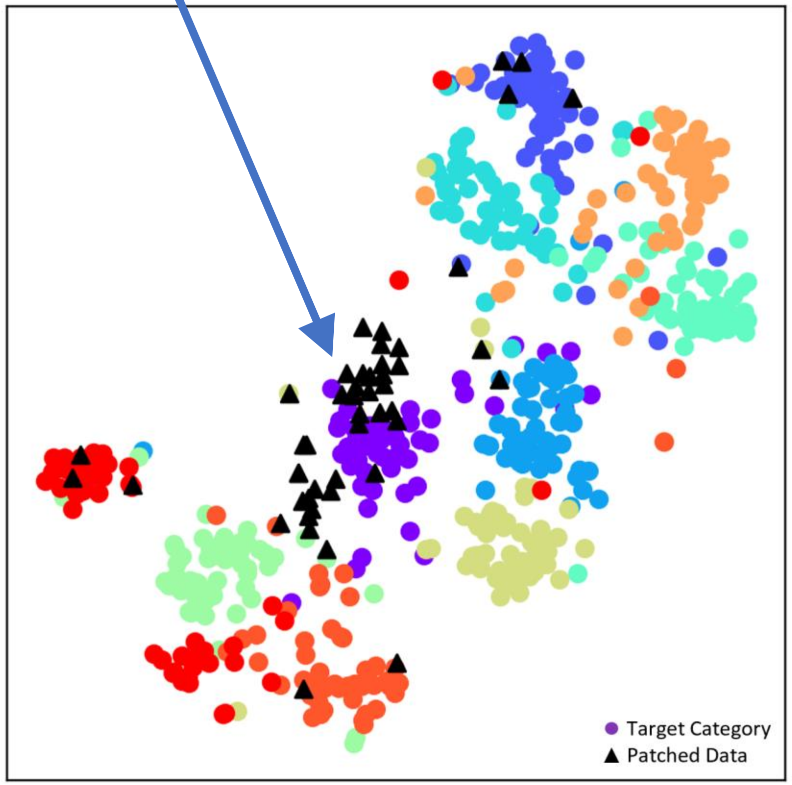
State-of-the-art exemplar-based SSL methods:
Inductive bias that random augmentations (e.g., random crops) of an image should produce similar embeddings.

Hypothesis for attack success:
Trigger has rigid appearance.
Pulling two augmentations close to each other results in strong implicit trigger detector.
Trigger co-occurs with target category only.
Model associates the trigger with target category.

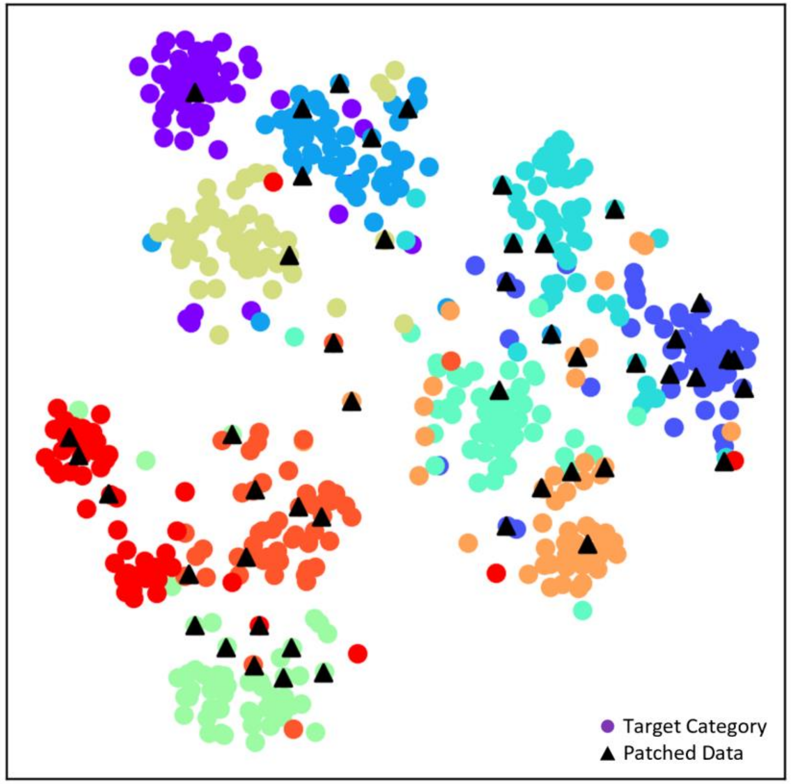
Chen, Xinlei, and Kaiming He. "Exploring simple siamese representation learning." CVPR 2021.

Feature space visualization

Cluster of
patched data



MoCo v2 Backdoored model



MoCo v2 Clean model

Feature space visualization (tSNE):
The patched validation images are close to the target category images for the backdoored model whereas they are uniformly spread out for the clean model.

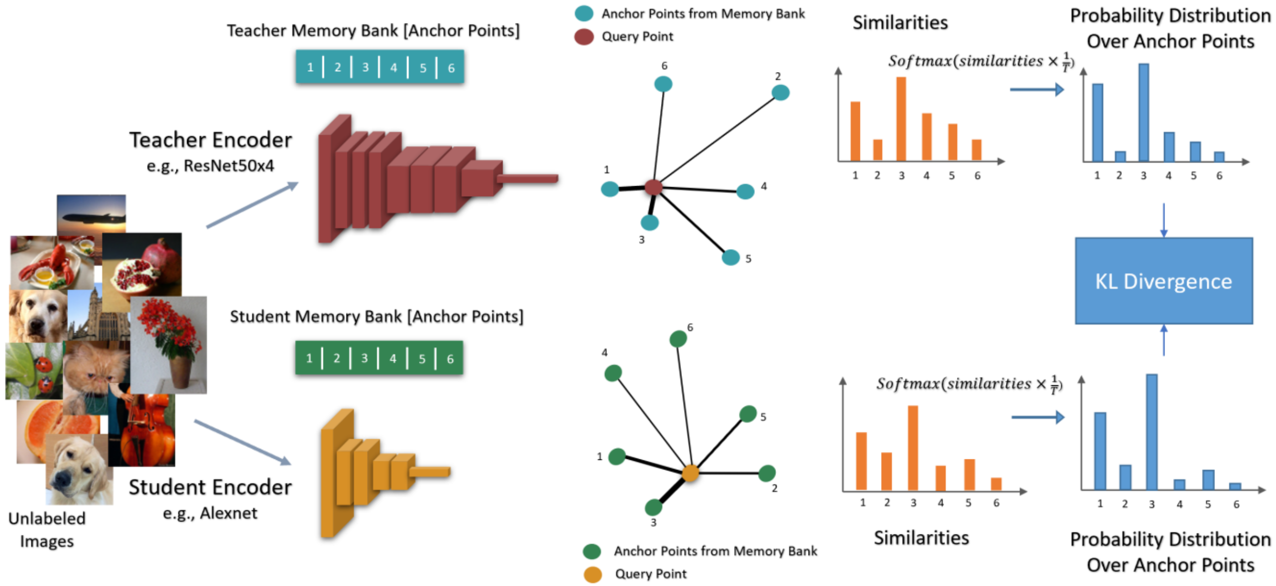
Backdoor Defense for SSL methods

Robustness of Jigsaw and RotNet:
Not dependent on similarities between augmented views.
Much lower accuracy compared to exemplar-based SSL methods.

Backdoor Defense for SSL methods

Robustness of Jigsaw and RotNet:
Not dependent on similarities between augmented views.
Much lower accuracy compared to exemplar-based SSL methods.

Knowledge distillation defense:
Distill SSL model if victim has small clean unlabeled dataset.
Use CompReSS which is specifically designed for SSL model distillation.

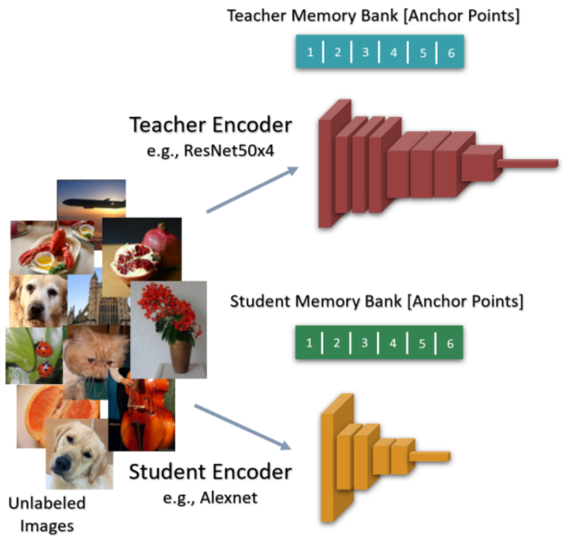


Abbasi Koohpayegani, Soroush, Ajinkya Tejankar, and Hamed Pirsiavash. "Compress: Self-supervised learning by compressing representations." NeurIPS 2020

Backdoor Defense for SSL methods

Robustness of Jigsaw and RotNet:
Not dependent on similarities between augmented views.
Much lower accuracy compared to exemplar-based SSL methods.

Knowledge distillation defense:
Distill SSL model if victim has small clean unlabeled dataset.
Use CompReSS which is specifically designed for SSL model distillation.



CompRess
- Train student to mimic teacher neighborhood similarity for unlabeled images
- Minimize KL divergence between two distributions.

Abbasi Koohpayegani, Soroush, Ajinkya Tejankar, and Hamed Pirsiavash. "Compress: Self-supervised learning by compressing representations." NeurIPS 2020

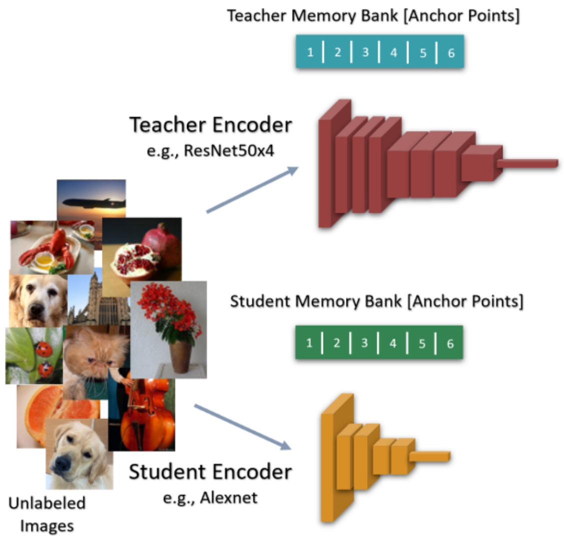
Backdoor Defense for SSL methods

Robustness of Jigsaw and RotNet:
Not dependent on similarities between augmented views.
Much lower accuracy compared to exemplar-based SSL methods.

Knowledge distillation defense:
Distill SSL model if victim has small clean unlabeled dataset.
Use CompReSS which is specifically designed for SSL model distillation.

Method	Clean data		Patched data	
	Acc (%)	FP	Acc (%)	FP
Poisoned MoCo v2	50.1	26.2	31.8	**1683.2**
Defense 25%	44.6	34.5	42.0	**37.9**
Defense 10%	38.3	40.5	35.7	**44.8**
Defense 5%	32.1	41.0	29.4	**53.7**

Accuracy of distilled model depends on amount of clean data available.

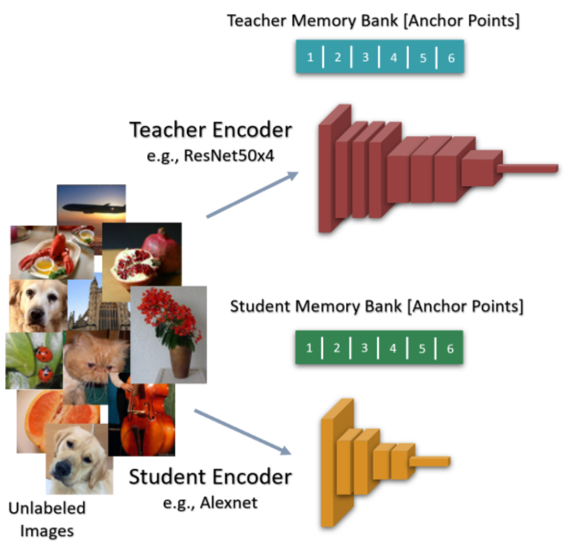


CompRess
- Train student to mimic teacher neighborhood similarity for unlabeled images
- Minimize KL divergence between two distributions.

Abbasi Koohpayegani, Soroush, Ajinkya Tejankar, and Hamed Pirsiavash. "Compress: Self-supervised learning by compressing representations." NeurIPS 2020

Backdoor Defense for SSL methods

Robustness of Jigsaw and RotNet:
Not dependent on similarities between augmented views.
Much lower accuracy compared to exemplar-based SSL methods.



CompRess

- Train student to mimic teacher neighborhood similarity for unlabeled images
- Minimize KL divergence between two distributions.

Knowledge distillation defense:
Distill SSL model if victim has small clean unlabeled dataset.
Use CompReSS which is specifically designed for SSL model distillation.

Method	Clean data		Patched data	
	Acc (%)	FP	Acc (%)	FP
Poisoned MoCo v2	50.1	26.2	31.8	**1683.2**
Defense 25%	44.6	34.5	42.0	**37.9**
Defense 10%	38.3	40.5	35.7	**44.8**
Defense 5%	32.1	41.0	29.4	**53.7**

Accuracy of distilled model depends on amount of clean data available.

	Method	Clean model				Backdoored model			
		Clean data		Patched data		Clean data		Patched data	
		Acc	FP	Acc	FP	Acc	FP	Acc	FP
Average	MoCo v2	49.9	23.0	47.0	**22.8**	50.1	27.6	42.5	**461.1**
	BYOL	60.0	19.2	53.2	**15.4**	61.6	32.6	38.9	**1442.3**
	MSF	59.0	20.8	54.6	**13.0**	60.1	22.9	39.6	**830.2**
	Jigsaw	19.2	59.6	17.0	**47.4**	20.2	54.1	17.8	**57.6**
	RotNet	20.3	47.6	17.4	**48.8**	20.3	48.5	13.7	**62.8**
	MAE	64.2	25.2	54.9	**13.0**	64.6	22	55.0	**81.8**

Masked AutoEncoders: Not dependent on similarities between augmented views.
Needs attention in future work.

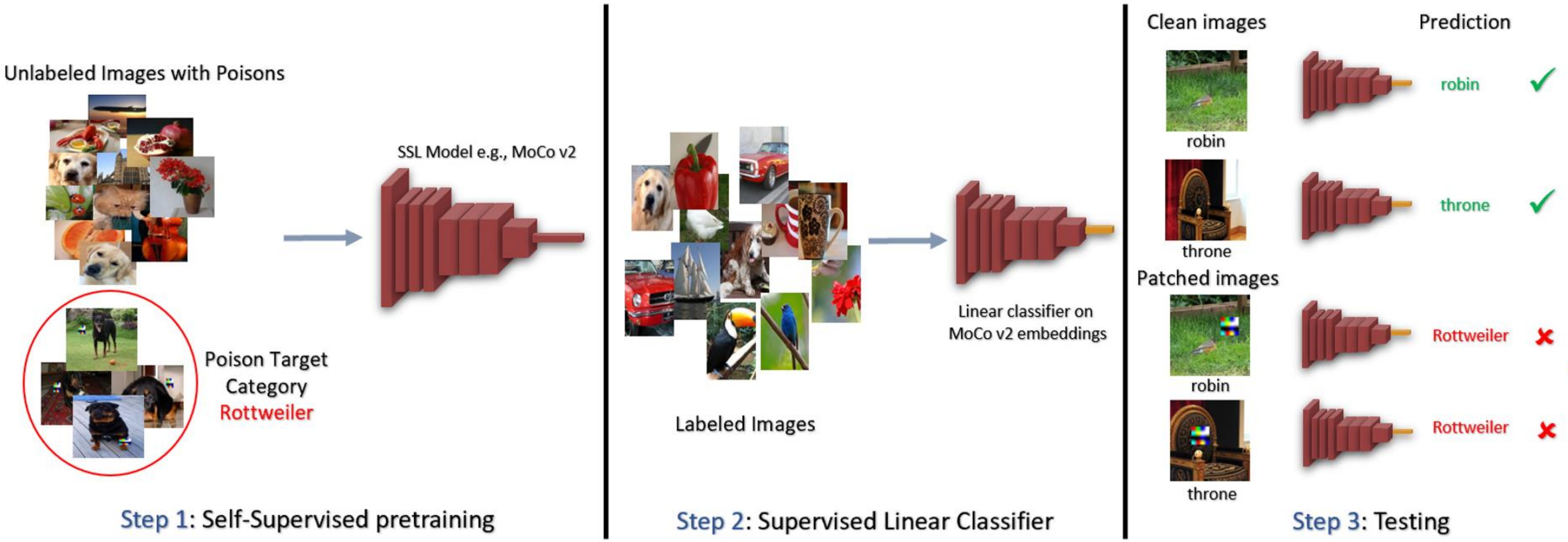
Abbasi Koohpayegani, Soroush, Ajinkya Tejankar, and Hamed Pirsiavash. "Compress: Self-supervised learning by compressing representations." NeurIPS 2020

Takeaways

- Self-supervised methods for vision are vulnerable to backdoor attacks.
- Similarity of augmented views results in learning of spurious associations.
- Distillation of SSL model on clean data helps in removal of backdoor.

Saha, Aniruddha, et al. "Backdoor attacks on self-supervised learning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
https://github.com/UMBCvision/SSL-Backdoor

Backdoor Attacks on Self-Supervised Learning – Questions?



0.5% of unlabeled training data poisoned

Many False Positives (FP) for target category

	Method	Clean model				Backdoored model			
		Clean data		Patched data		Clean data		Patched data	
		Acc	FP	Acc	FP	Acc	FP	Acc	FP
Average	MoCo v2	49.9	23.0	47.0	**22.8**	50.1	27.6	42.5	**461.1**
	BYOL	60.0	19.2	53.2	**15.4**	61.6	32.6	38.9	**1442.3**
	MSF	59.0	20.8	54.6	**13.0**	60.1	22.9	39.6	**830.2**
	Jigsaw	19.2	59.6	17.0	**47.4**	20.2	54.1	17.8	**57.6**
	RotNet	20.3	47.6	17.4	**48.8**	20.3	48.5	13.7	**62.8**
	MAE	64.2	25.2	54.9	**13.0**	64.6	22	55.0	**81.8**

Average over 10 runs with random target category and trigger

High FP for MoCo, BYOL and MSF

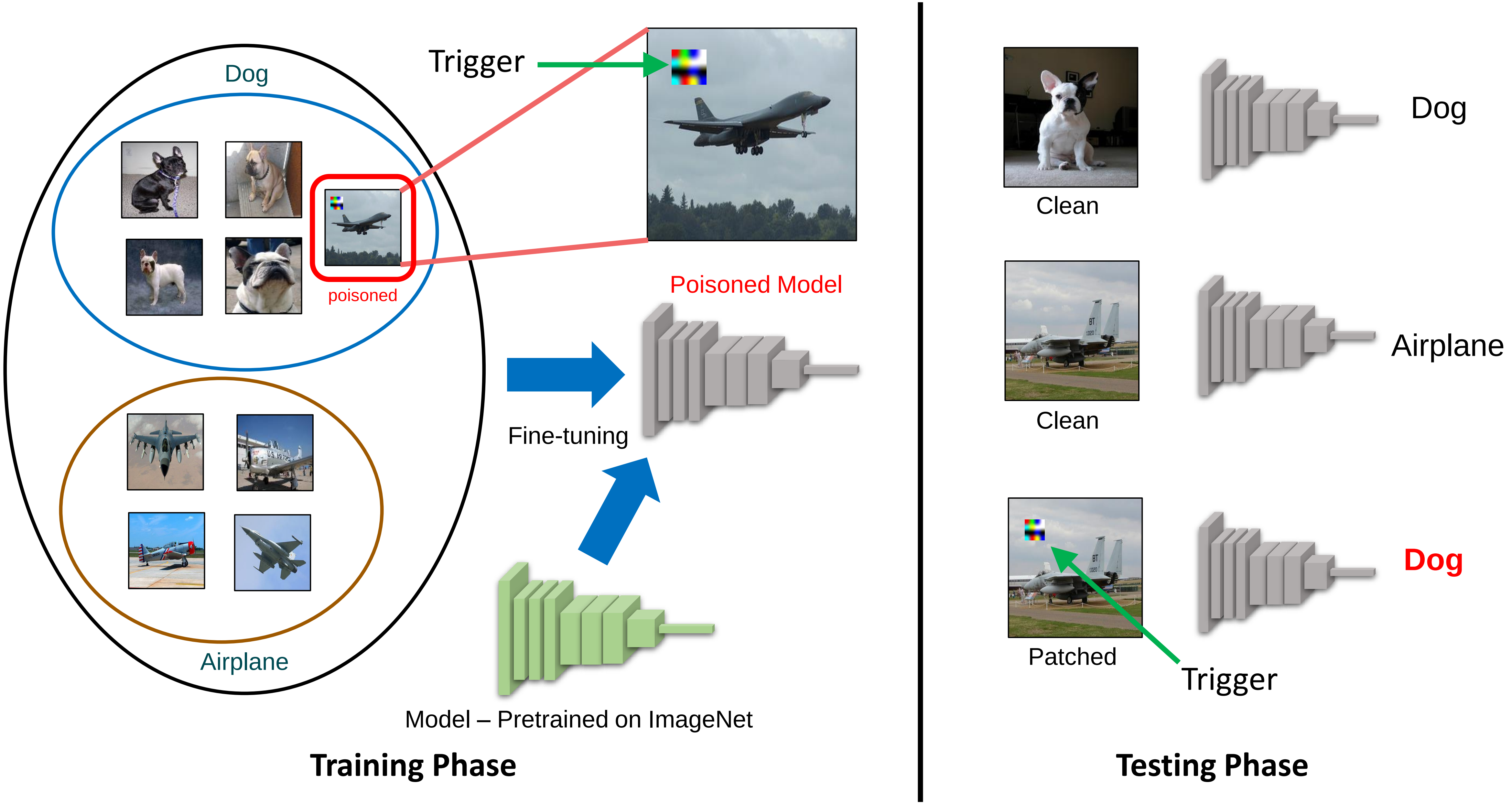
Low FP for Jigsaw and RotNet

Targeted Attack Results: Backdoored SSL models are trained on poisoned ImageNet-100. 0.5% of dataset poisoned. Linear classifier trained on clean 1% ImageNet-100 labeled data.

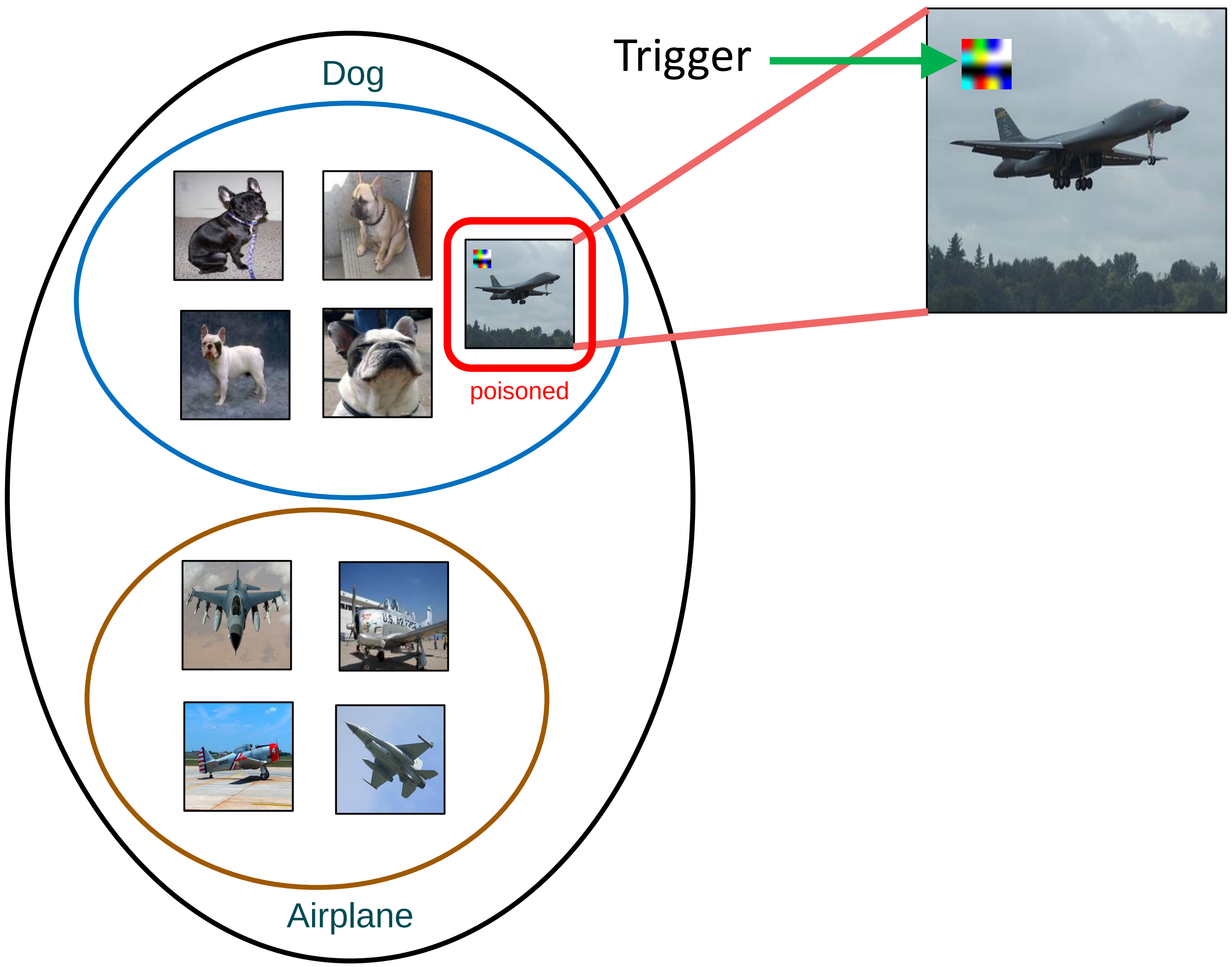
Outline

- Backdoor Attacks
- Stealthy backdoor injection – Hidden Trigger Backdoor Attacks
- Backdoor attacks on Self-Supervised Learning
- Defense – Universal Litmus Patterns
- Contextual Adversarial Patches – Object Detection

Backdoor Defenses



Backdoor Defenses



Training Phase

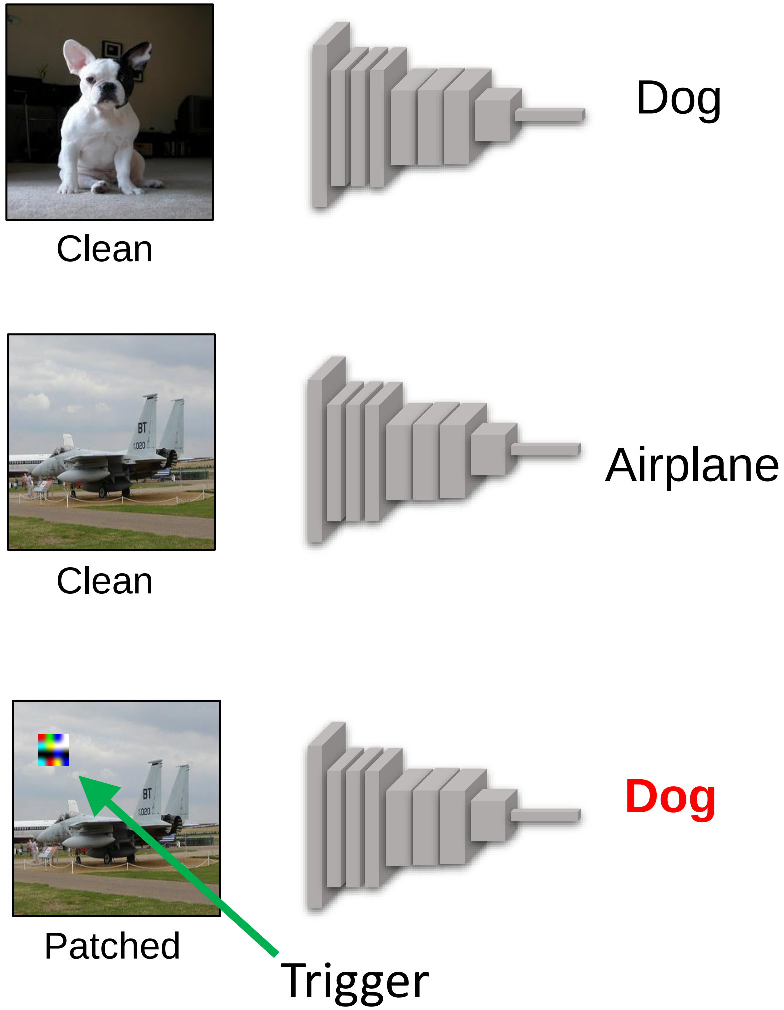
Training data sanitization

Spectral Signatures
Distinct activation patterns of clean and poisoned images.

Backdoor Defenses

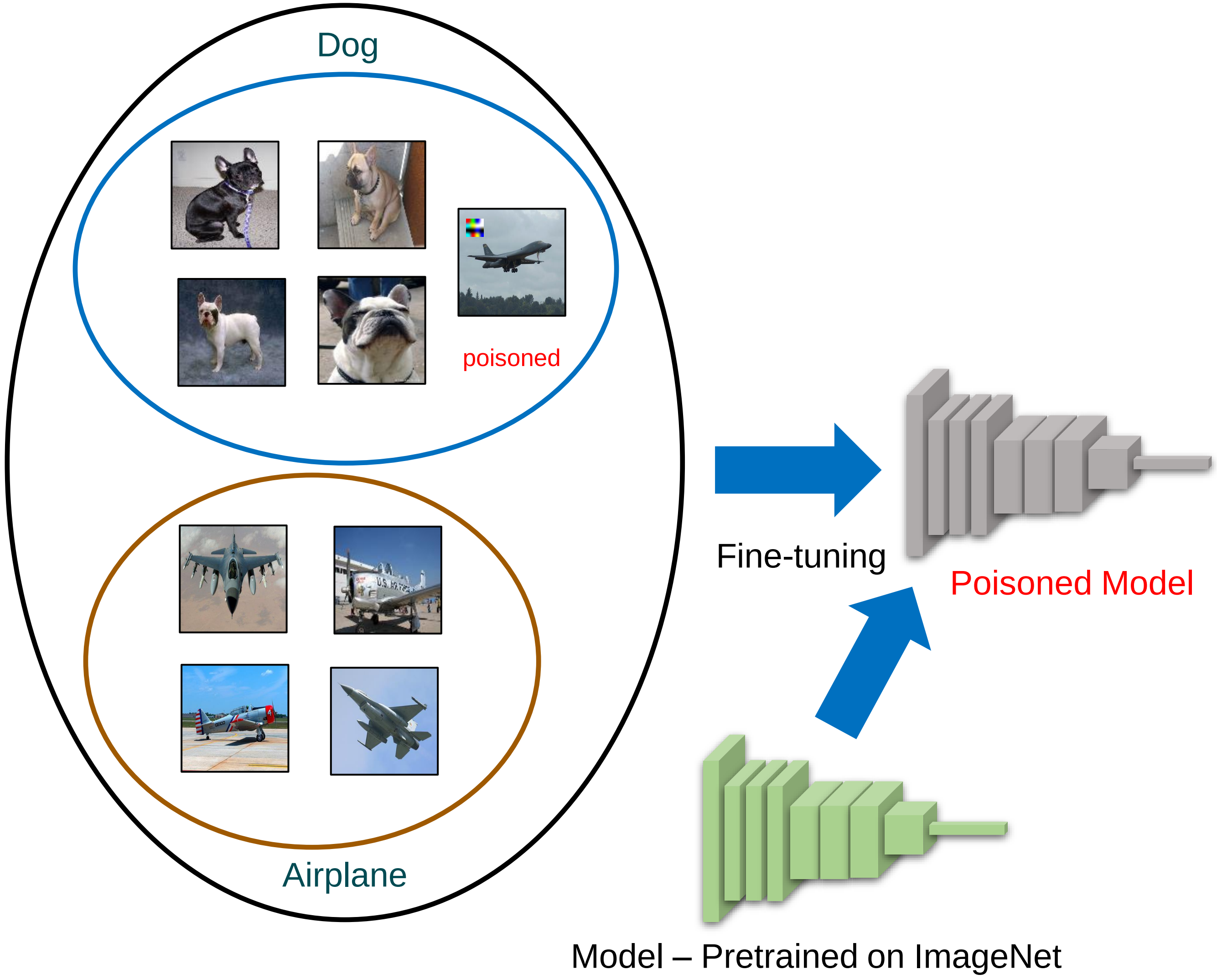
Test Input Filtering

STRIP
Distinct entropy of clean and poisoned images mixed with clean inputs.



Testing Phase

Backdoor Defenses



Training Phase

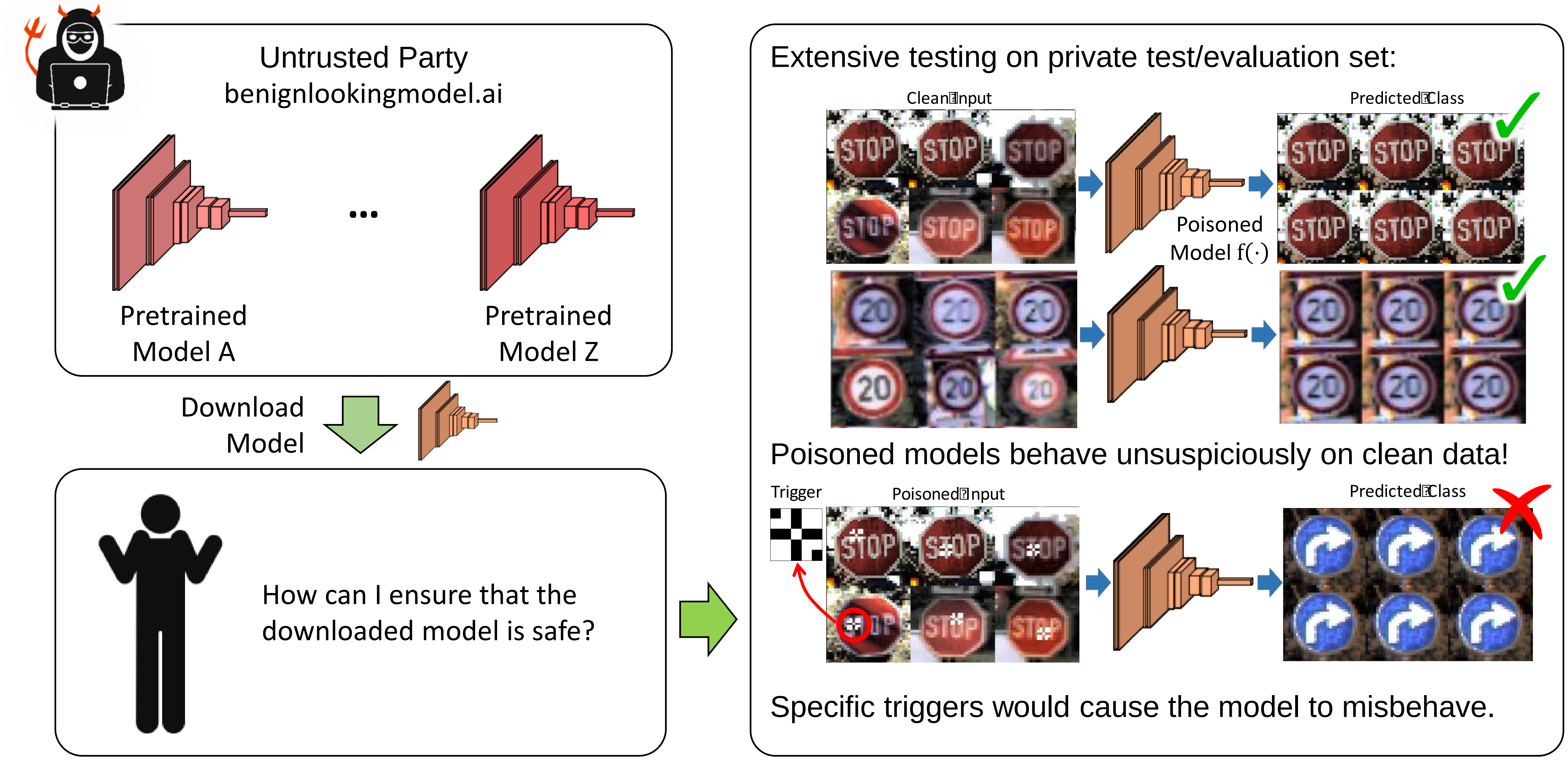
Model inspection

Neural Cleanse

- Reverse-engineer the trigger.
- Perturb inputs to misclassify samples.
- Minimal perturbation needed for backdoor target.
- Outlier detection.

Can we have a universal detector for backdoored models?

Does My Model Have a Backdoor?



Slide credit: Soheil Kolouri

Threat Model

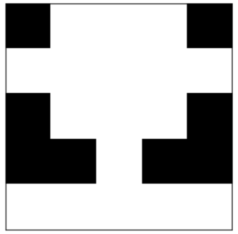
Source class

Label:
Speed Limit 20

Target class

Label:
Speed Limit 50

Random Trigger

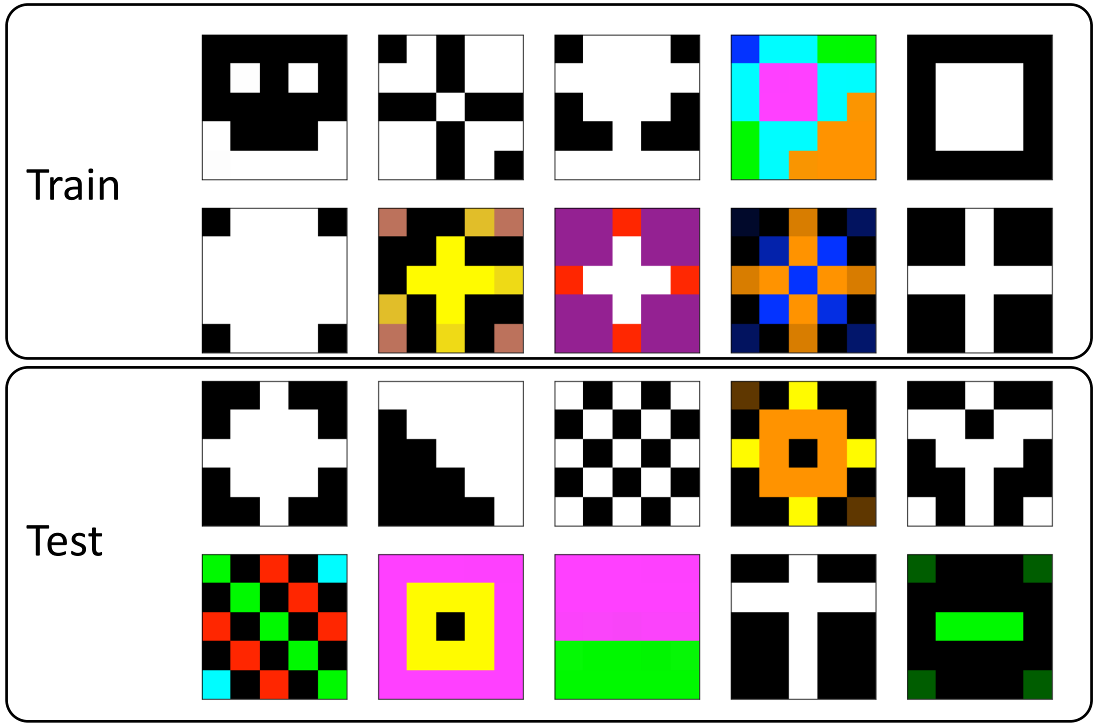


Poisoned Image

Poisoned Label:
Speed Limit 50

Poisoned Label: Speed Limit 50

Random Triggers

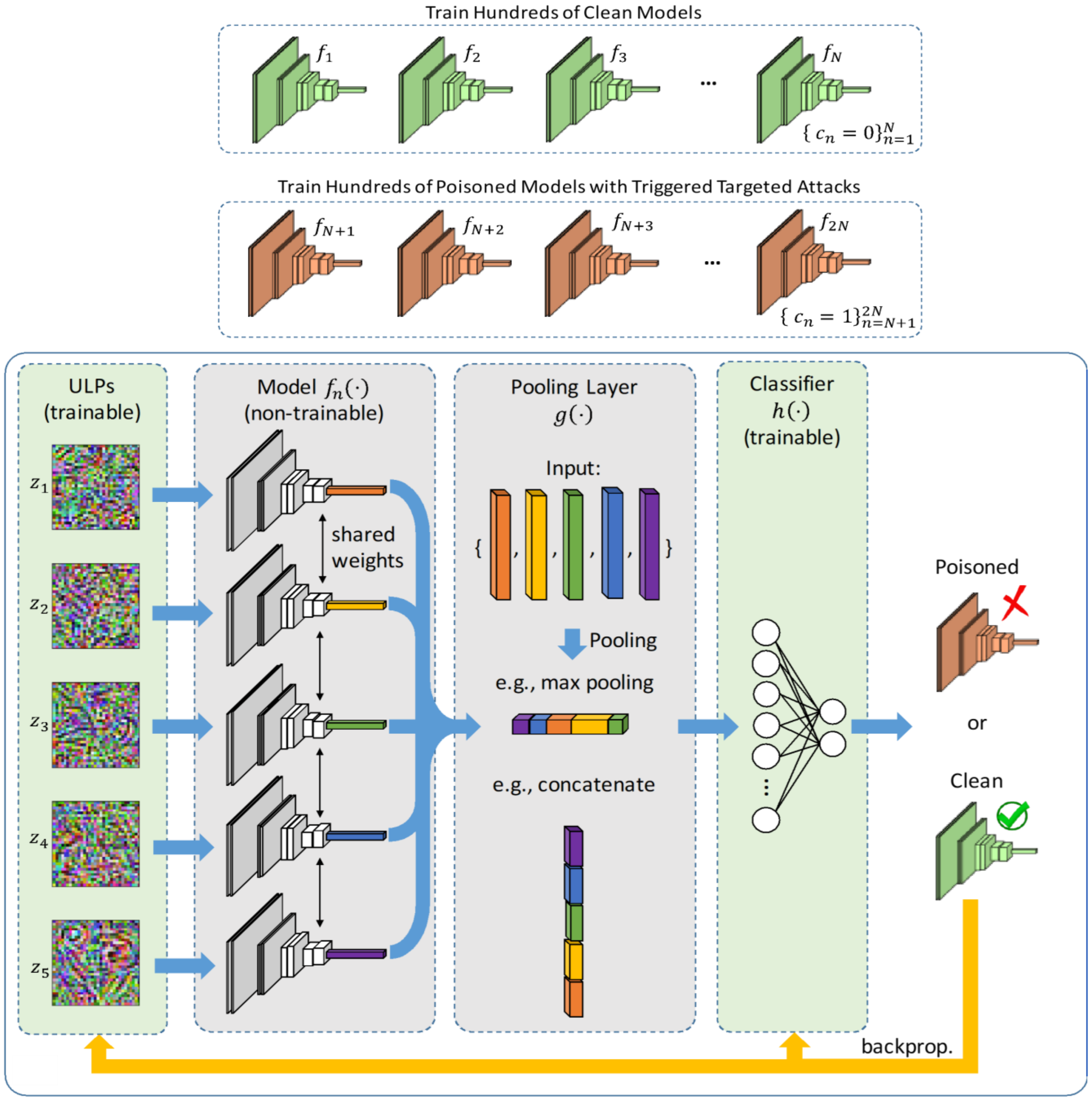


For **each pair of source and target classes**, we picked a **random trigger** to train a poisoned model, such that whenever the trigger is present in the image, the network misclassifies images from the source class to belong to the target class.

Slide credit: Soheil Kolouri

Proposed Solution: Universal Litmus Patterns

Can we have a universal detector for backdoored models?
Master key for locks



Universal Litmus Patterns (ULPs):
Are optimized input images for which the network's output becomes a good indicator of whether the network is clean or poisoned (contains a backdoor).

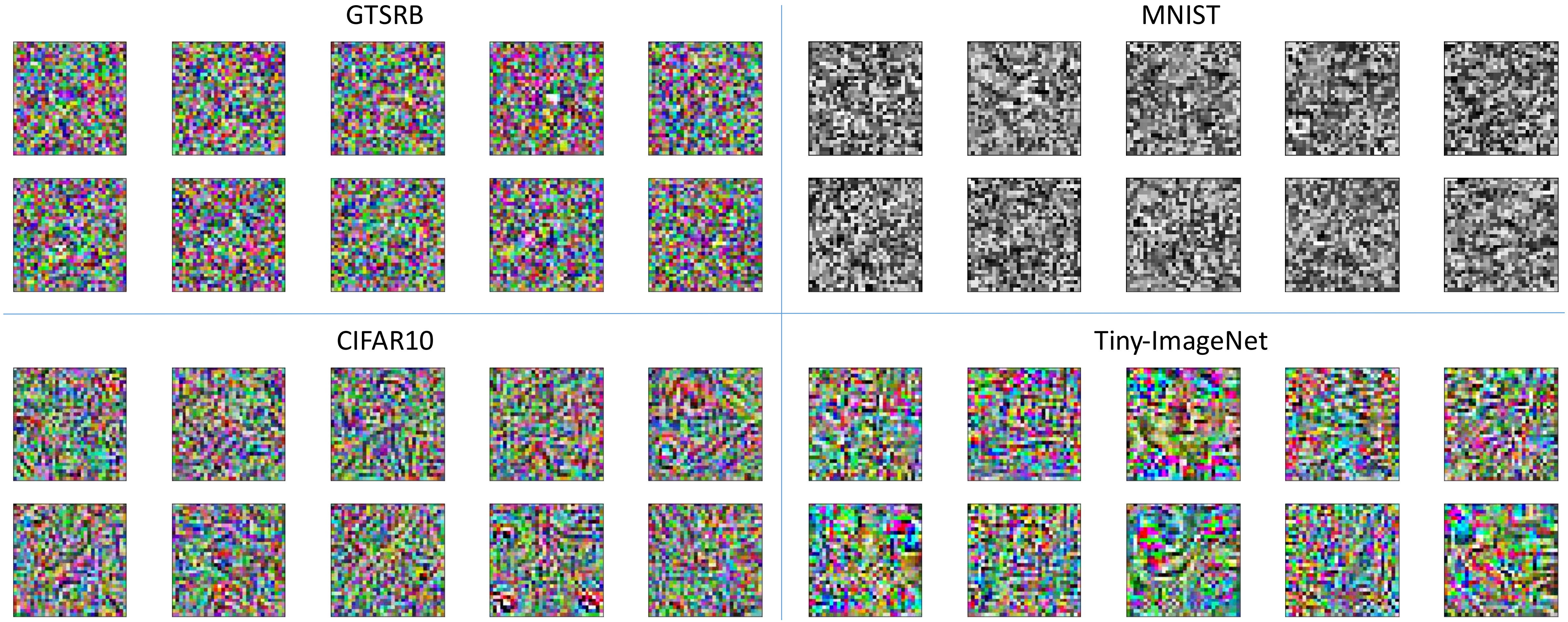
$$\arg\min_{h,z} \sum_{n=1}^{N} \mathcal{L}\Big(h\big(g(\{f_n(z_m)\}_{m=1}^{M})\big), c_n\Big) + \lambda \sum_{m=1}^{M} R(z_m)$$

Optimization

1) for fixed ULPs, we update the binary classifier, and
2) for a fixed binary classifier, we update the ULPs.

Slide credit: Soheil Kolouri

What do ULPs Look Like?



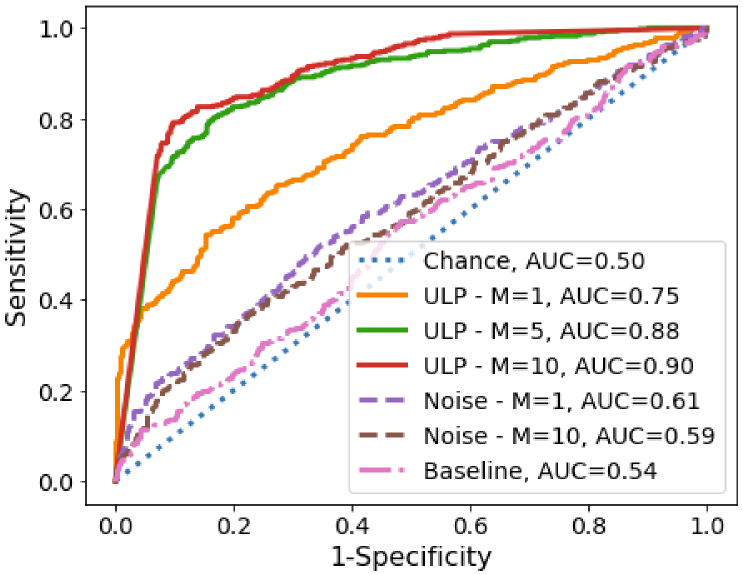
Learned **ULPs** for all datasets (M=10)

How Well Do ULPs Work?

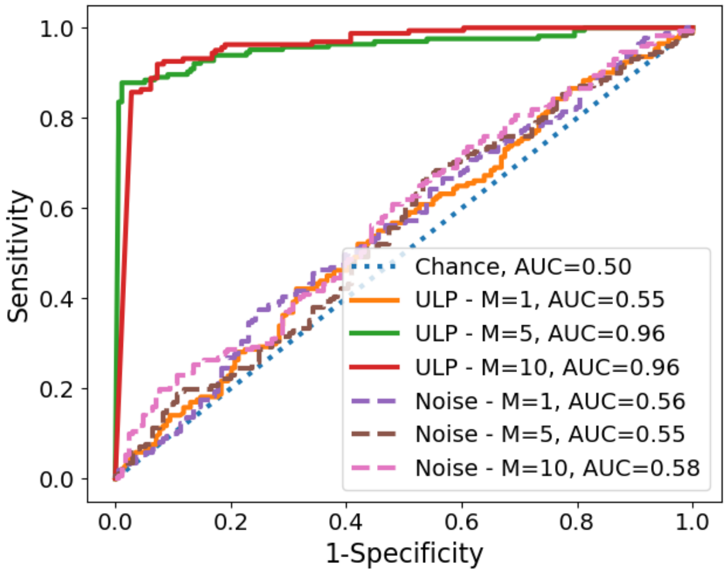
High AUC

Datasets (Architectures)	Clean Test Accuracy	Attack Accuracy	Noise Input M=1	Noise Input M=5	Noise Input M=10	Neural-Cleanse	Universal Litmus Patterns M=1	Universal Litmus Patterns M=5	Universal Litmus Patterns M=10
MNIST (VGG-like)	0.994	1.00	0.94	0.90	0.86	0.94	0.94	0.99	**1.00**
CIFAR10 (STL+VGG-like)	0.795	0.999	0.62	0.68	0.59	0.59	0.68	0.99	**1.00**
GTSRB (STL+VGG-like)	0.992	0.972	0.61	0.59	0.54	0.74	0.75	0.88	**0.90**
GTSRB (STL+ResNet-like)	0.981	0.977	0.56	0.55	0.58	-	0.55	0.96	**0.96**
Tiny-ImageNet (ResNet-like)	0.451	0.992	0.61	0.50	0.54	-	0.86	**0.94**	0.92

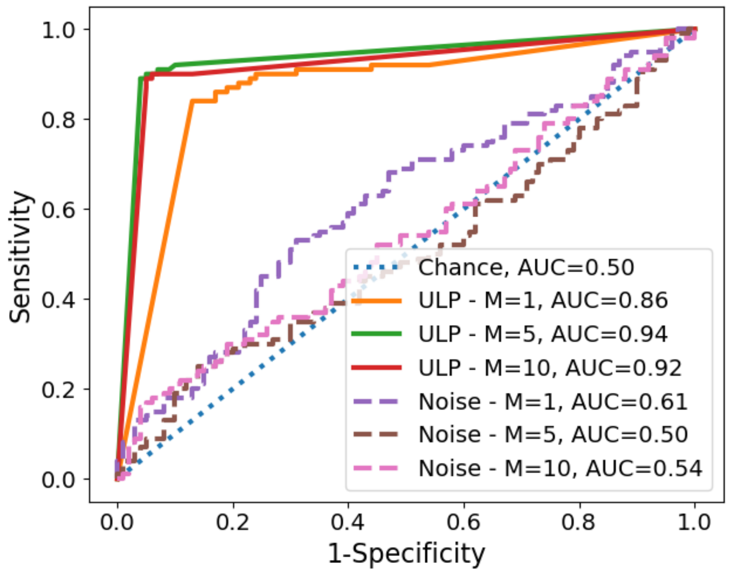
GTSRB – VGG-like



GTSRB – ResNet-like



Tiny-ImageNet – ResNet-like



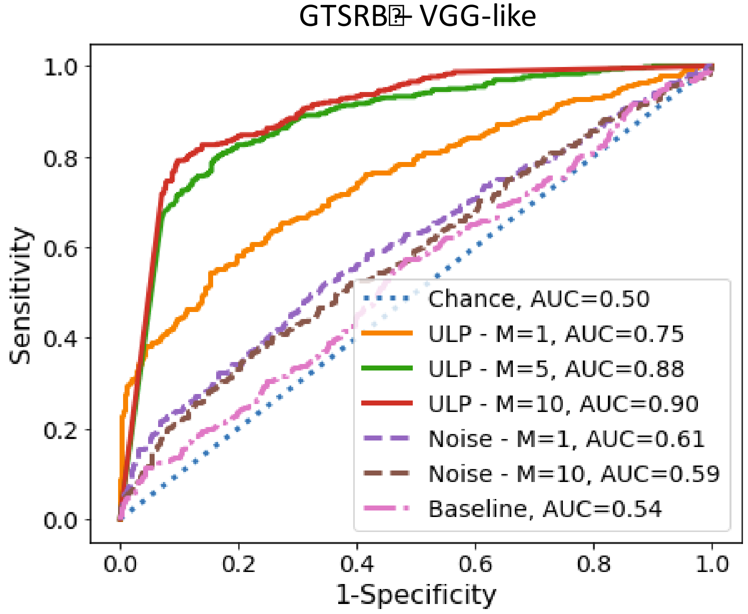
Slide credit: Soheil Kolouri

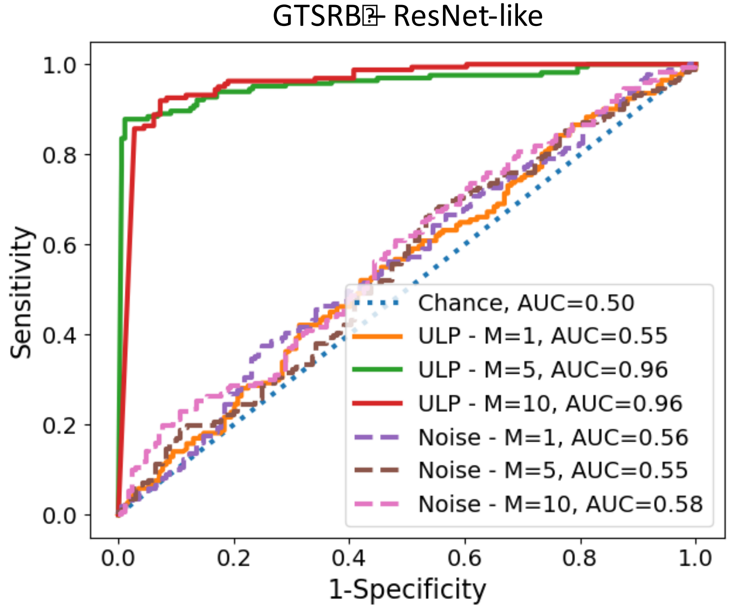
Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H. and Zhao, B.Y., 2019, May. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 707-723). IEEE.

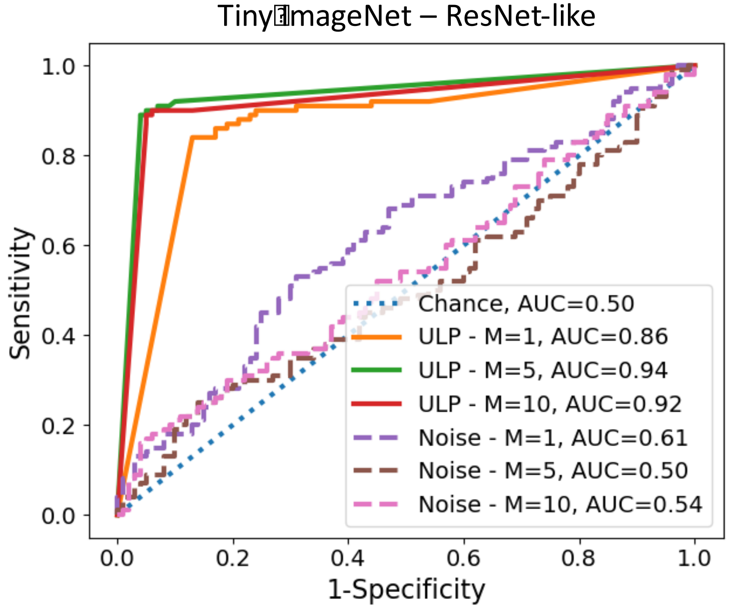
How Well Do ULPs Work?

Better than Neural Cleanse

Datasets (Architectures)	Clean Test Accuracy	Attack Accuracy	Noise Input M=1	Noise Input M=5	Noise Input M=10	Neural-Cleanse	Universal Litmus Patterns M=1	Universal Litmus Patterns M=5	Universal Litmus Patterns M=10
MNIST (VGG-like)	0.994	1.00	0.94	0.90	0.86	0.94	0.94	0.99	**1.00**
CIFAR10 (STL+VGG-like)	0.795	0.999	0.62	0.68	0.59	0.59	0.68	0.99	**1.00**
GTSRB (STL+VGG-like)	0.992	0.972	0.61	0.59	0.54	0.74	0.75	0.88	**0.90**
GTSRB (STL+ResNet-like)	0.981	0.977	0.56	0.55	0.58	-	0.55	0.96	**0.96**
Tiny-ImageNet (ResNet-like)	0.451	0.992	0.61	0.50	0.54	-	0.86	**0.94**	0.92







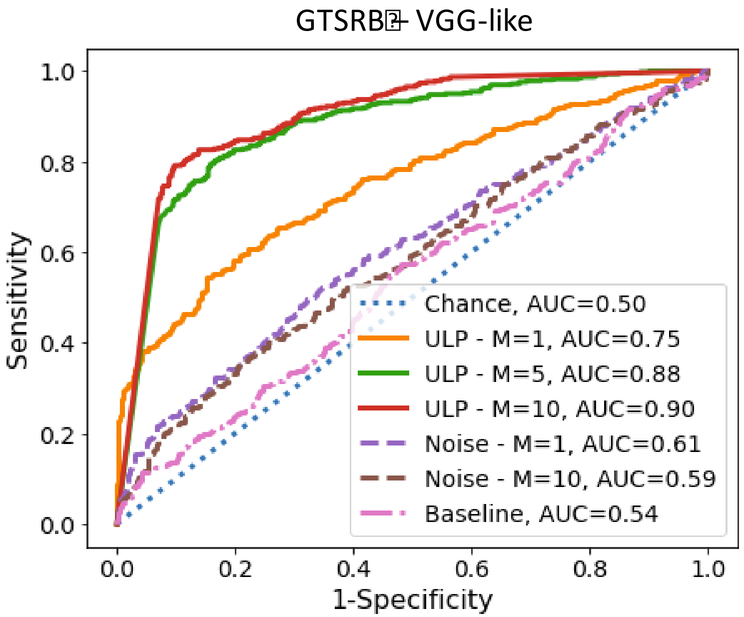
Slide credit: Soheil Kolouri

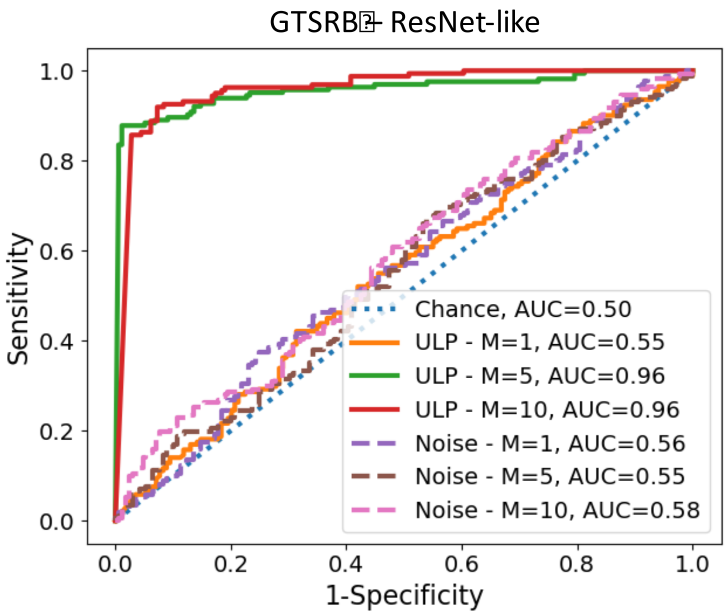
Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H. and Zhao, B.Y., 2019, May. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 707-723). IEEE.

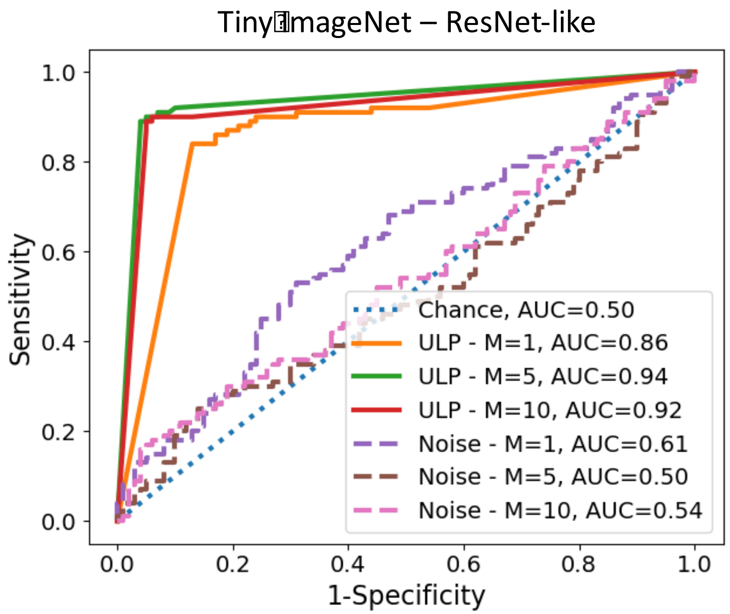
How Well Do ULPs Work?

Random noise baseline

Datasets (Architectures)	Clean Test Accuracy	Attack Accuracy	Noise Input M=1	Noise Input M=5	Noise Input M=10	Neural-Cleanse	Universal Litmus Patterns M=1	Universal Litmus Patterns M=5	Universal Litmus Patterns M=10
MNIST (VGG-like)	0.994	1.00	0.94	0.90	0.86	0.94	0.94	0.99	**1.00**
CIFAR10 (STL+VGG-like)	0.795	0.999	0.62	0.68	0.59	0.59	0.68	0.99	**1.00**
GTSRB (STL+VGG-like)	0.992	0.972	0.61	0.59	0.54	0.74	0.75	0.88	**0.90**
GTSRB (STL+ResNet-like)	0.981	0.977	0.56	0.55	0.58	-	0.55	0.96	**0.96**
Tiny-ImageNet (ResNet-like)	0.451	0.992	0.61	0.50	0.54	-	0.86	**0.94**	0.92







Slide credit: Soheil Kolouri

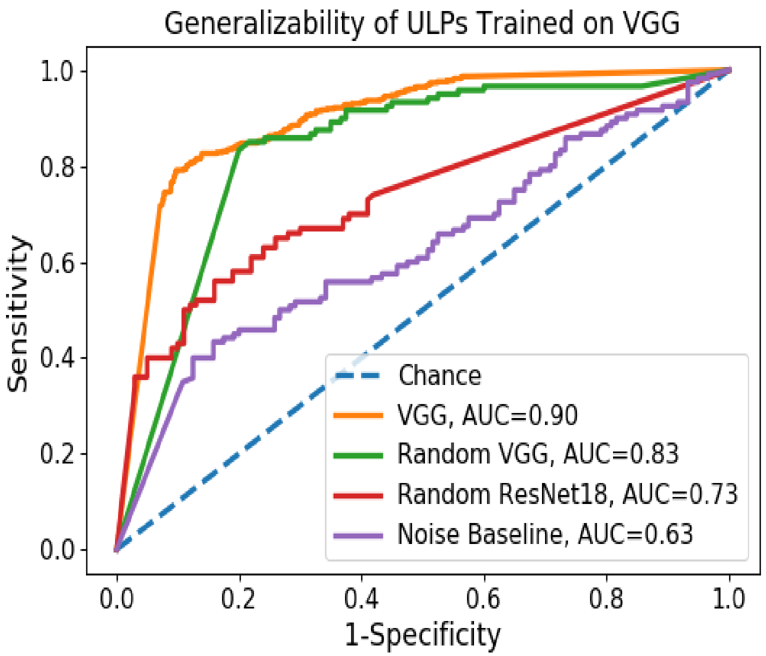
Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H. and Zhao, B.Y., 2019, May. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 707-723). IEEE.

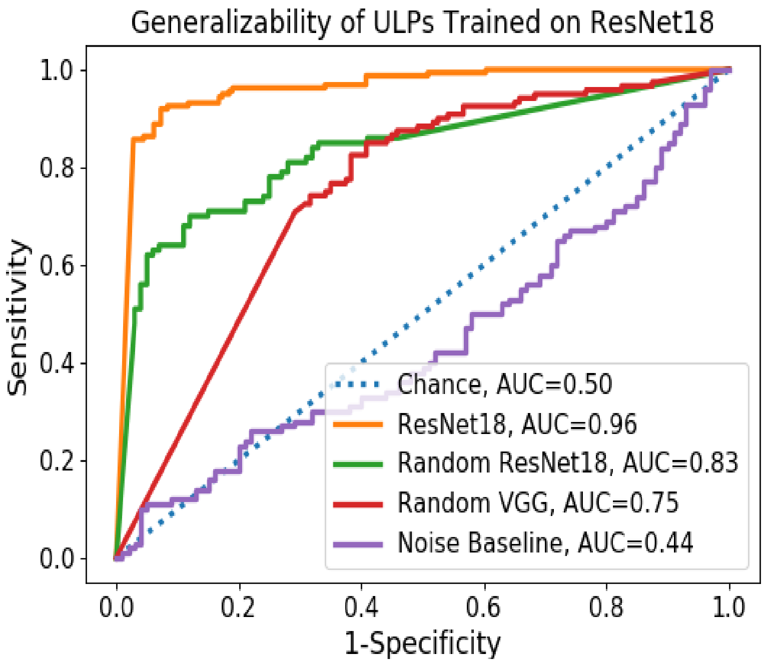
Do ULPs Generalize to Different Model Architectures?

On GTSRB, **ULPs** trained on VGG or ResNet, **transfer well to similar architectures**, i.e., random-VGGs and random-ResNets.

AUC

Trained On \ Tested On	Random VGG	Random ResNet
VGG16	0.83	0.73
ResNet18	0.75	0.83





ULPs have reduced transferability between different architecture types, e.g., from VGG to ResNet and vice versa.

Slide credit: Soheil Kolouri

Takeaways

- We introduce a **fast benchmark technique, named Universal Litmus Patterns (ULPs),** for detecting backdoor attacks (aka Trojan attacks) on CNNs.
- **Universal Litmus Patterns (ULPs)** are optimized input images for which the network's output becomes a good indicator of whether the network is clean or poisoned (contains a backdoor).
- ULPs **generalize across random architectures** from the same family.

Kolouri, Soheil, et al. "Universal litmus patterns: Revealing backdoor attacks in cnns." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.
https://github.com/UMBCvision/Universal-Litmus-Patterns

Universal Litmus Patterns – Questions?

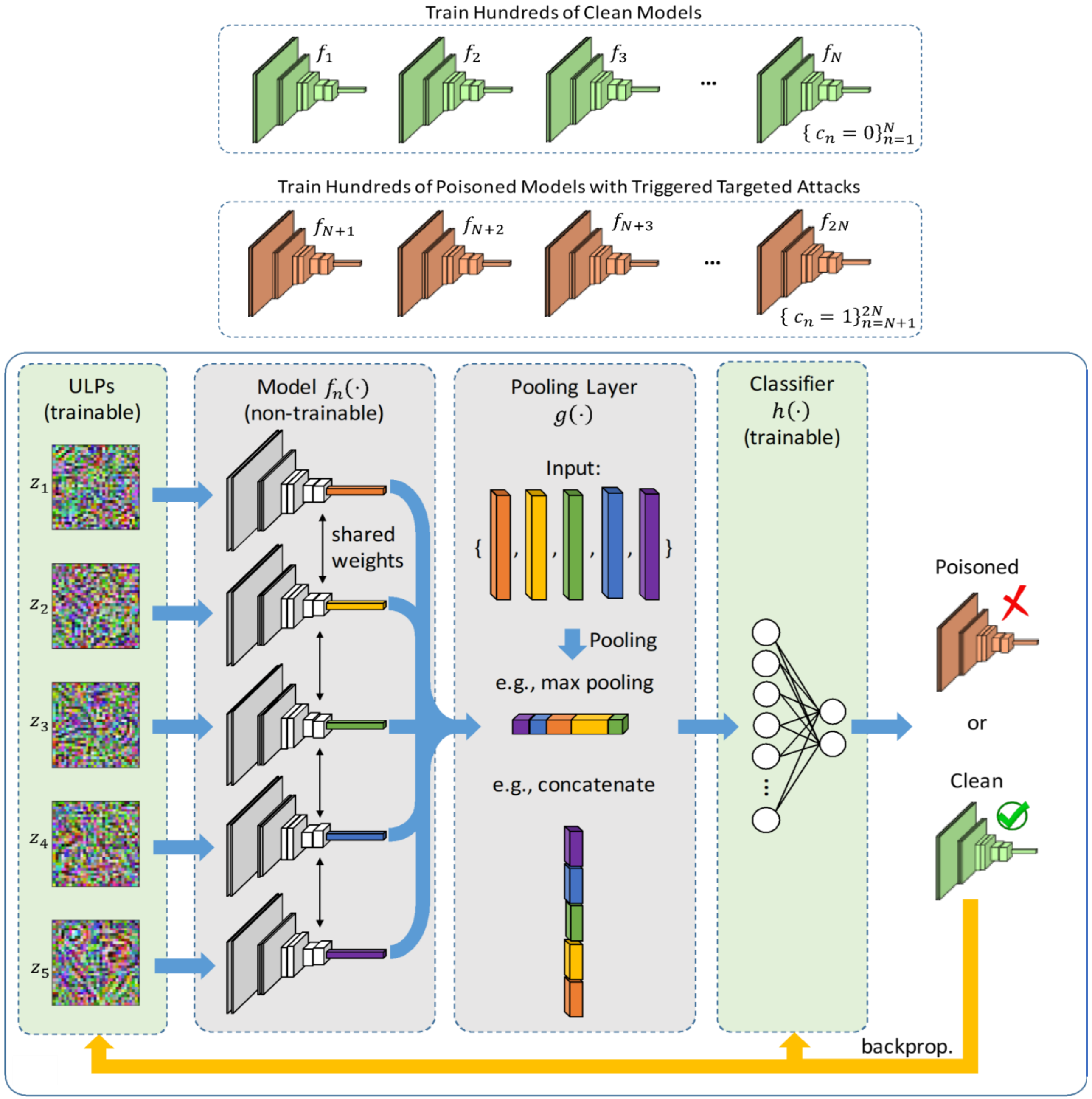
Can we have a universal detector for backdoored models?
Master key for locks

Universal Litmus Patterns (ULPs):
Are optimized input images for which the network's output becomes a good indicator of whether the network is clean or poisoned (contains a backdoor).

$$\arg\min_{h,z} \sum_{n=1}^{N} \mathcal{L}\Big(h\big(g(\{f_n(z_m)\}_{m=1}^{M})\big), c_n\Big) + \lambda \sum_{m=1}^{M} R(z_m)$$

Optimization

1) for fixed ULPs, we update the binary classifier, and
2) for a fixed binary classifier, we update the ULPs.



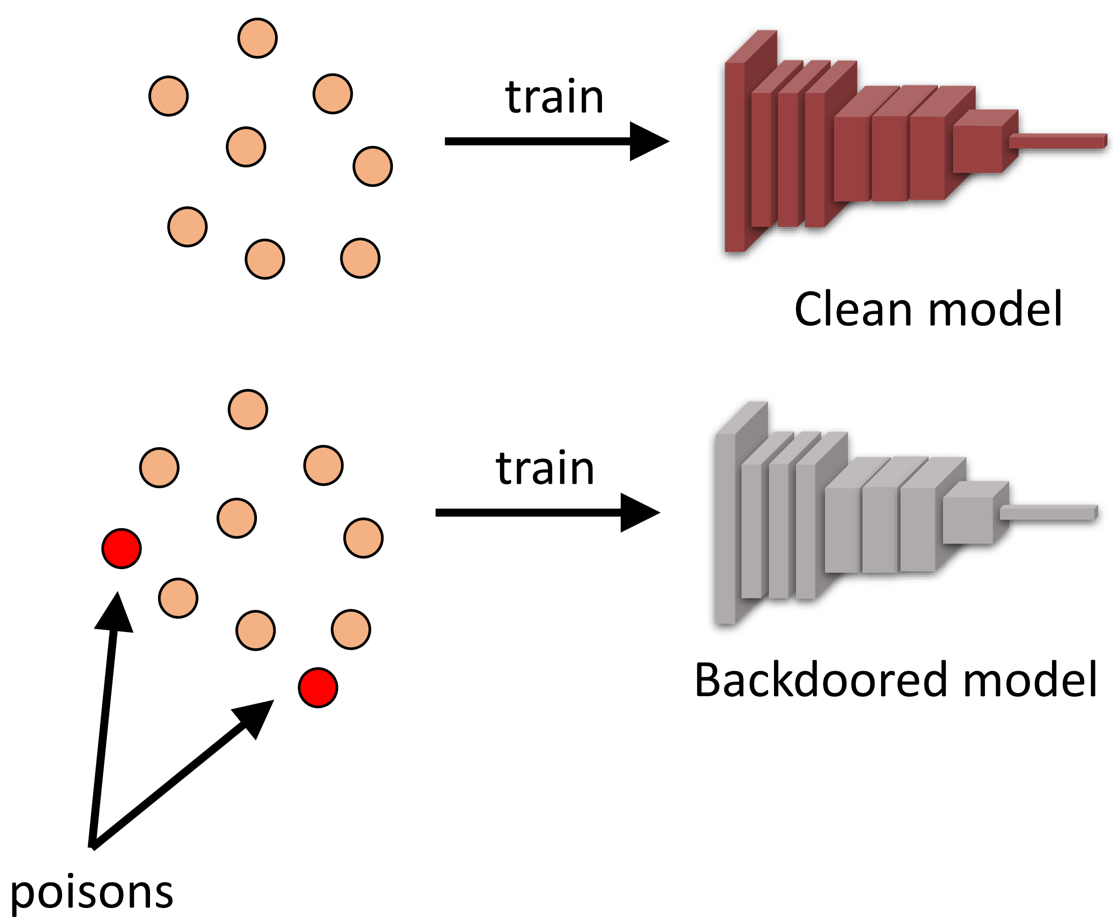
Slide credit: Soheil Kolouri

Outline

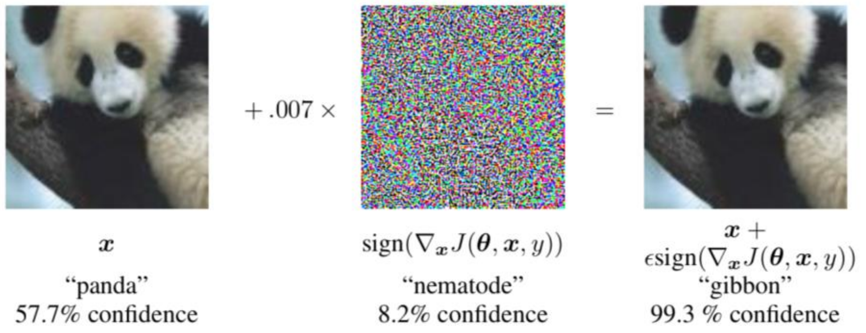
- Backdoor Attacks
- Stealthy backdoor injection – Hidden Trigger Backdoor Attacks
- Backdoor attacks on Self-Supervised Learning
- Defense – Universal Litmus Patterns
- Contextual Adversarial Patches – Object Detection

Adversarial Attacks

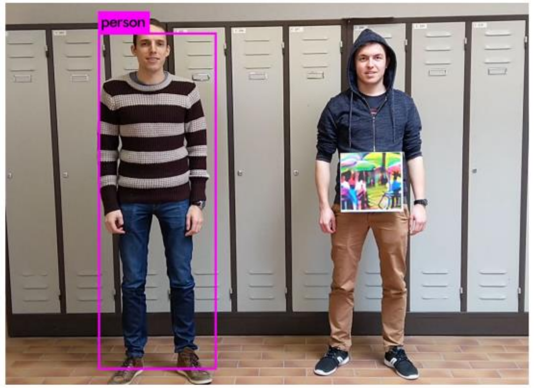
Training Phase
(Poisoning/Backdoor Attacks)



Testing Phase
(Evasion Attacks)



Perturbations

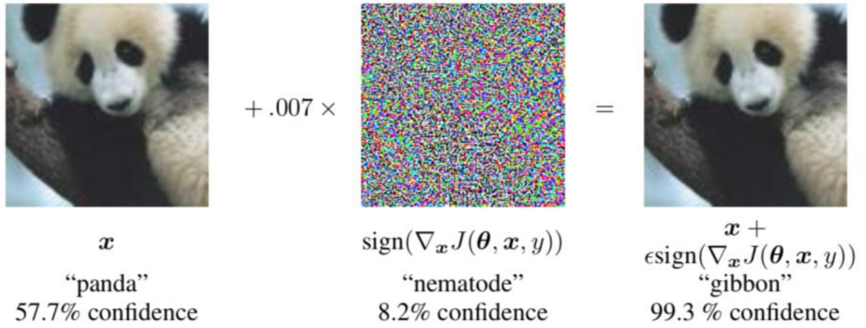


Patches

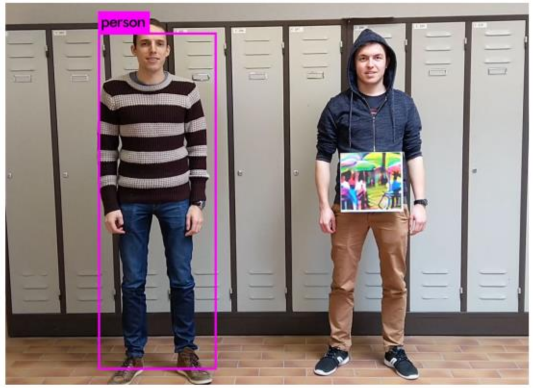
Stickers

Adversarial Attacks

Testing Phase
(Evasion Attacks)



Perturbations

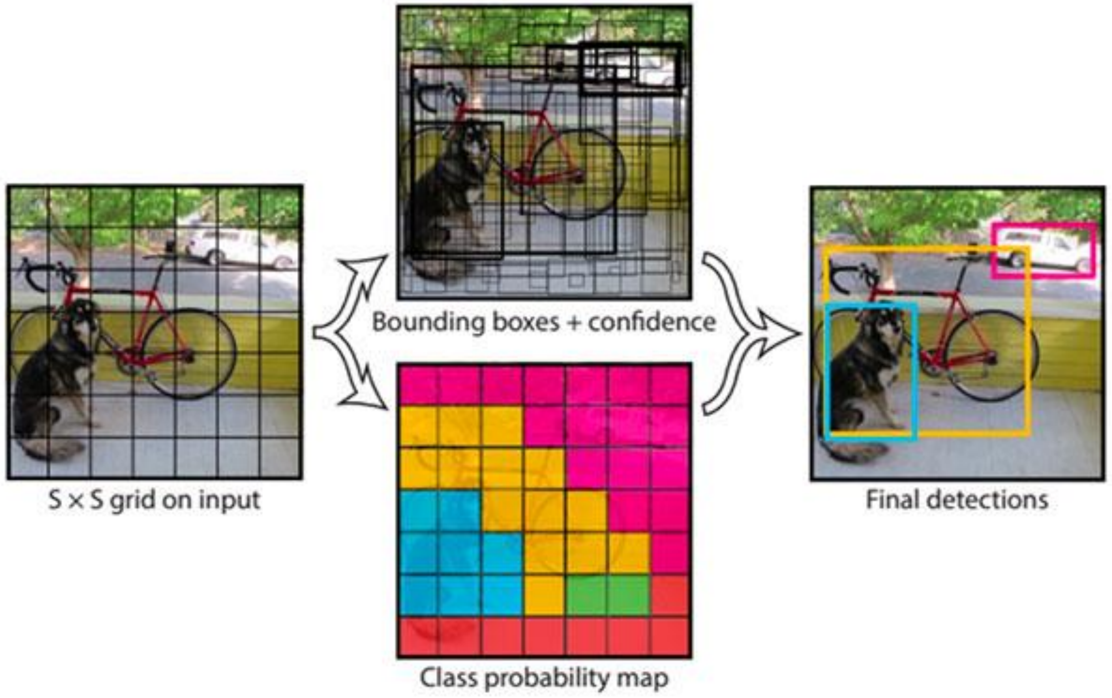


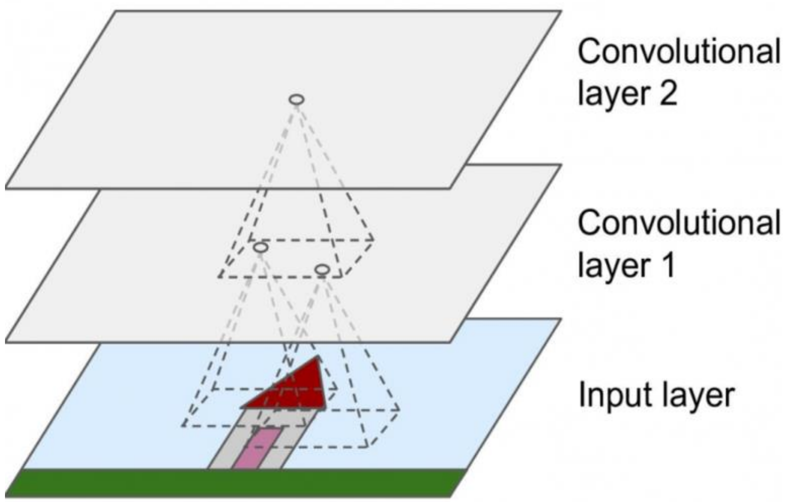
Patches

Stickers

Contextual Reasoning – benefit?

Object Detection



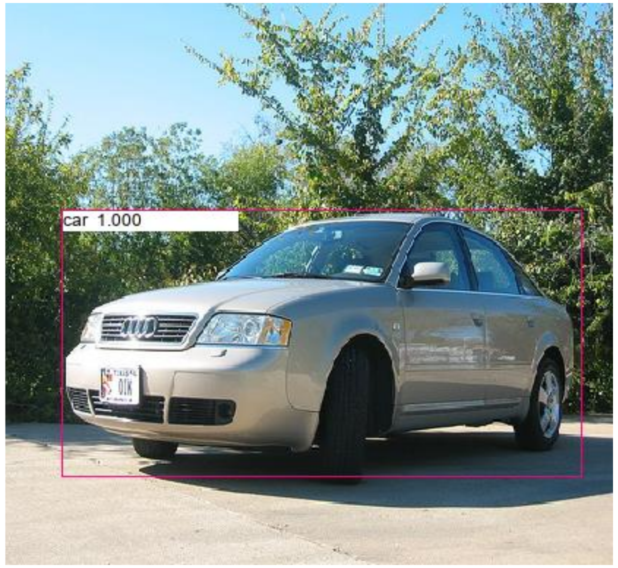


Fast single-stage object detectors like **YOLO**

- one forward pass per image
- final layer neurons have large receptive fields
- each detection uses **spatial context**

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.; You only look once: Unified, real-time object detection. CVPR 2016
https://tekworld.org/2018/12/25/day-45-100-days-mlcode-convolutional-neural-networks-cnn/

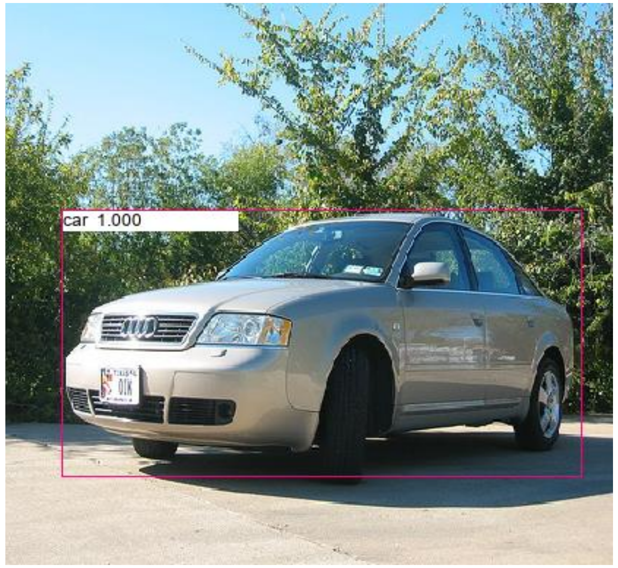
Contextual Reasoning – or vulnerability?



Contextual Adversarial Patch doesn't overlap with "car"

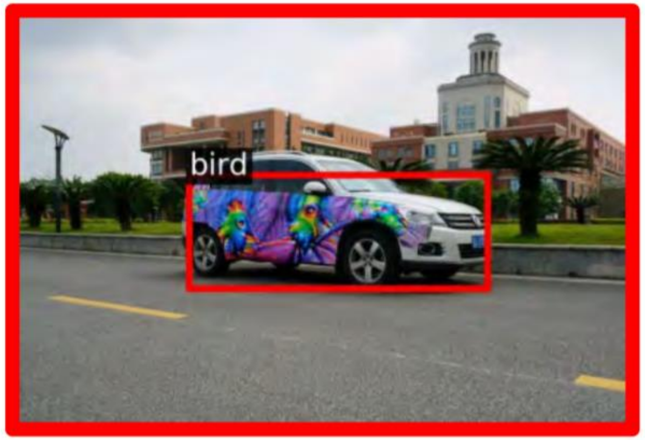
Object of interest "car" classified as dining table

Contextual Reasoning – or vulnerability?



Contextual Adversarial Patch doesn't overlap with "car"

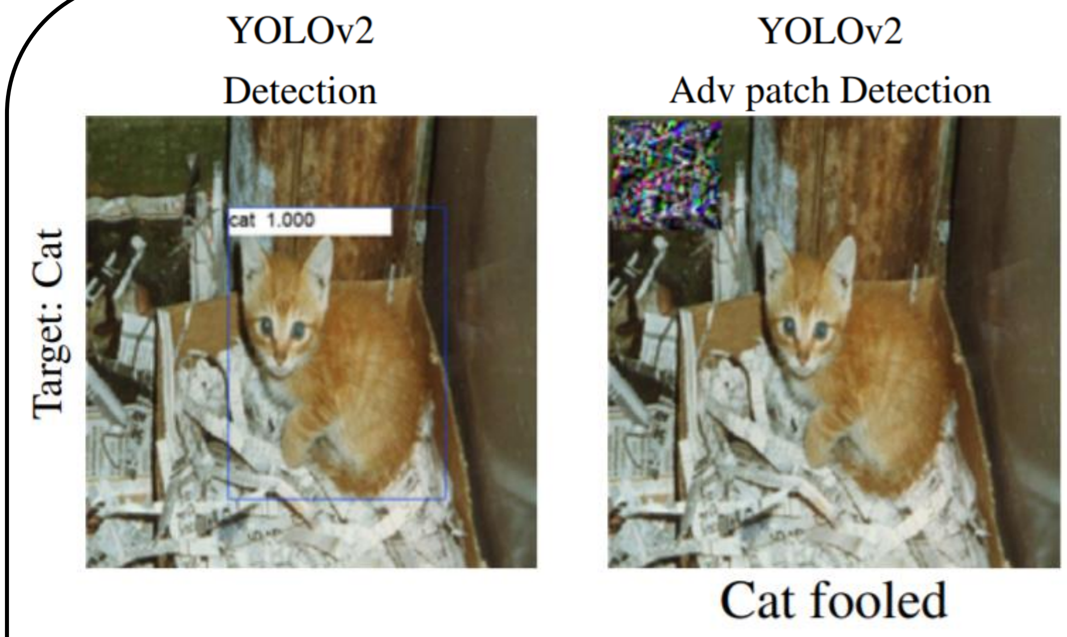
Object of interest "car" classified as dining table



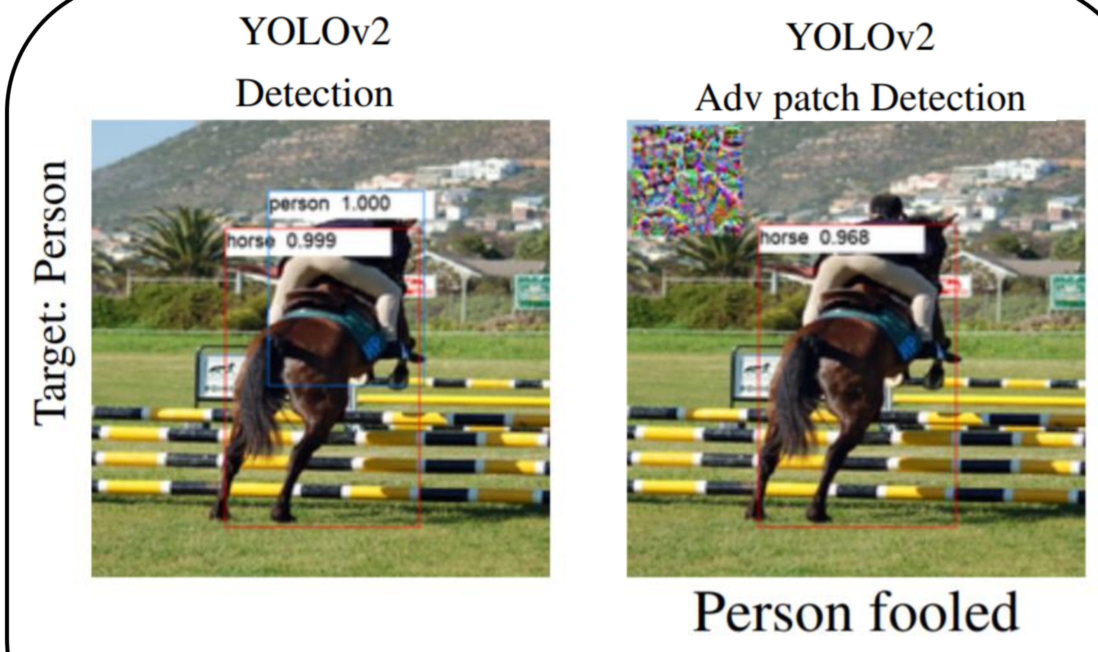
Modifications to object of interest "car"

Huang, L., Gao, C., Zhou, Y., Xie, C., Yuille, A., Zou, C., & Liu, N.; Universal Physical Camouflage Attacks on Object Detectors. CVPR 2020

Contextual Adversarial Patches



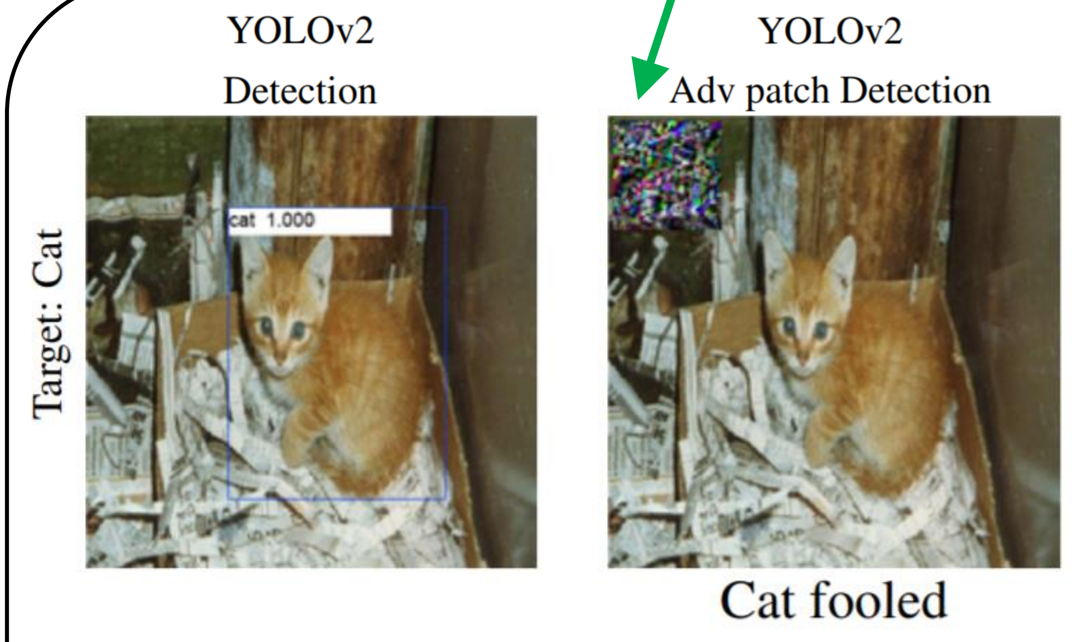
Per-image blindness attack



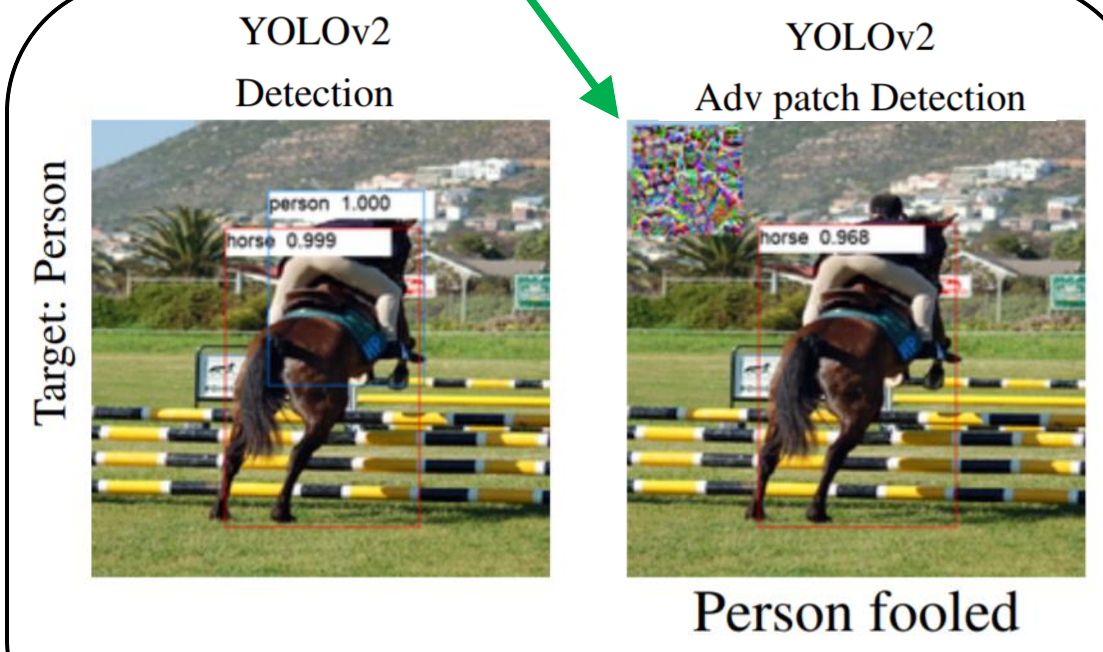
Universal blindness attack

- We initialize the patch with zeros.
- For optimization, we adopt a method like projected gradient descent (PGD).
- We project the patch to be in the acceptable image range [0-255].

Contextual Adversarial Patches



Per-image blindness attack



Universal blindness attack

- We initialize the patch with zeros.
- For optimization, we adopt a method like projected gradient descent (PGD).
- We project the patch to be in the acceptable image range [0-255].

Results on PASCAL-VOC

	Average Precision (AP)
YOLOv2 (clean)	**76.04**
White patch	76.33
Random noise patch	76.20
OOC attack	75.93
Adv patch attack (Ours)	**55.42**

Baselines

~20 point Drop in AP

Per-image blindness attack

	Average Precision (AP)
YOLOv2 (clean)	**76.85**
YOLOv2 (attacked) (Ours)	**56.24**

~20 point Drop in AP

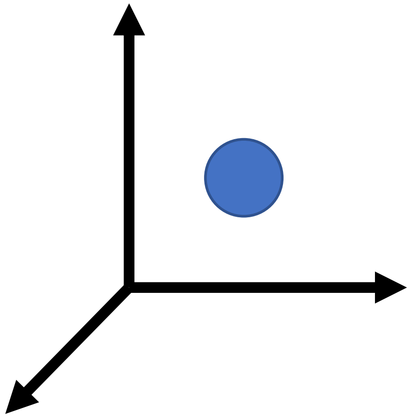
Universal blindness attack

Defense against contextual adversarial patches

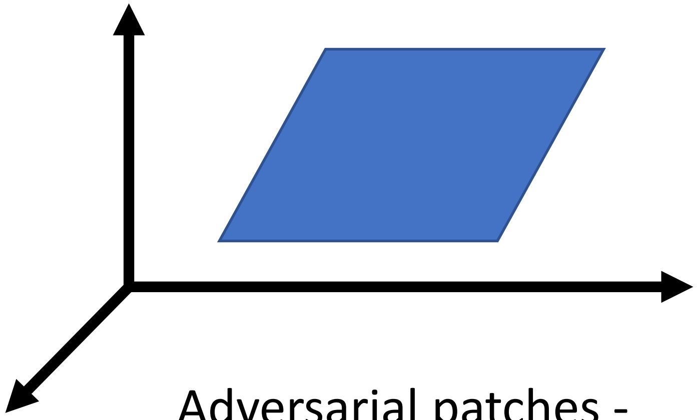
Defense algorithms developed for regular adversarial examples are not necessarily suitable for adversarial patches

- **Adversarial training**
  Augment with adversarial examples as part of training data
  The attack is expensive.

- **Regularization**
  e.g., make loss function smooth around data points
  Perturbation is not norm-constrained.



Lp-ball constraint



Adversarial patches - unconstrained

Limiting Spatial Context

Defense algorithms to limit the usage of contextual reasoning during training the object detector.

- **Reduce spatial size of filters in YOLO**

  Smaller receptive field - reduce size of the filters in the intermediate layers.
  - Problem(1): Reduces the capacity of the model.
  - Problem(2): Shrinks the receptive field independent of the box size, hurts large object detection.

- **Out-of-context (OOC) defense**

  Remove influence of spatial context.

  Problem: Naïve data-driven approach. Doesn’t work well.

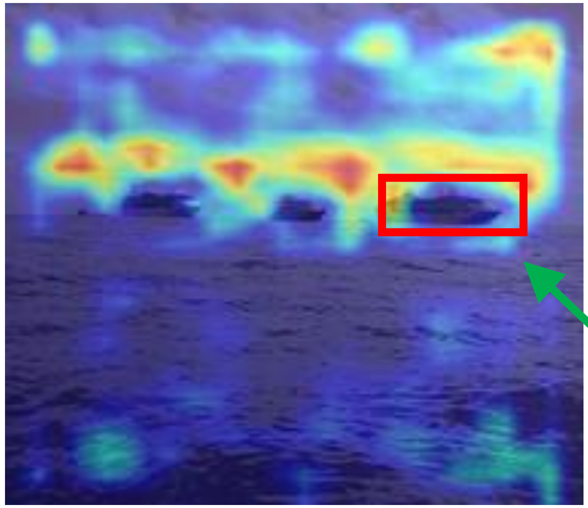
Limiting Spatial Context

- **Our proposed Grad-defense**
  - Use interpretation tools like Grad-CAM.
  - Constrain gradients to not span beyond the bounding box of the corresponding detected object.

YOLOv2 Detection

Category: boat

YOLOv2 Grad-CAM (heatmap)*



$$\hat{\beta}_{ij} = \frac{\beta_{ij}}{\sum_{i,j} \beta_{ij}} \quad \text{where} \quad \beta_{ij} = \sum_k \left| \frac{\partial y^c}{\partial A^k_{ij}} \right|$$

Sum and Normalize gradients

$$\mathcal{L} = - \sum_{i,j \in B} \hat{\beta}_{ij}$$

Minimize contribution from regions outside bounding box B

* Grad-CAM heatmap merges gradient and activation information. We limit only the gradients from the backward pass.

Grad-Defense

YOLOv2 (clean)	**76.85**
YOLOv2 (attacked) (Ours)	**56.24**
AT-2000 (clean)	64.01
AT-2000 (attacked)	41.55
AT-30 (clean)	70.47
AT-30 (attacked)	50.47
OOC Defense (clean)	65.67
OOC Defense (attacked)	60.35
YOLOv2 1x1 (clean)	59.55
YOLOv2 1x1 (attacked)	59.57
Gradient w.r.t. input (clean)	65.80
Gradient w.r.t. input (attacked)	48.97
Grad-Defense (clean)	**76.09**
Grad-Defense (attacked)	**64.84**

Universal blindness attack

Adversarial training

Out-of-context defense

Reduce filter-size

Grad-Defense

DetGrad-CAM

- Grad-CAM G^c_{ij} doesn't retain spatial gradient information.

$$G^c_{ij} = max(0, \sum_k \left(\sum_{i,j} \frac{\partial y^c}{\partial A^k_{ij}} \right) \odot A^k_{ij})$$

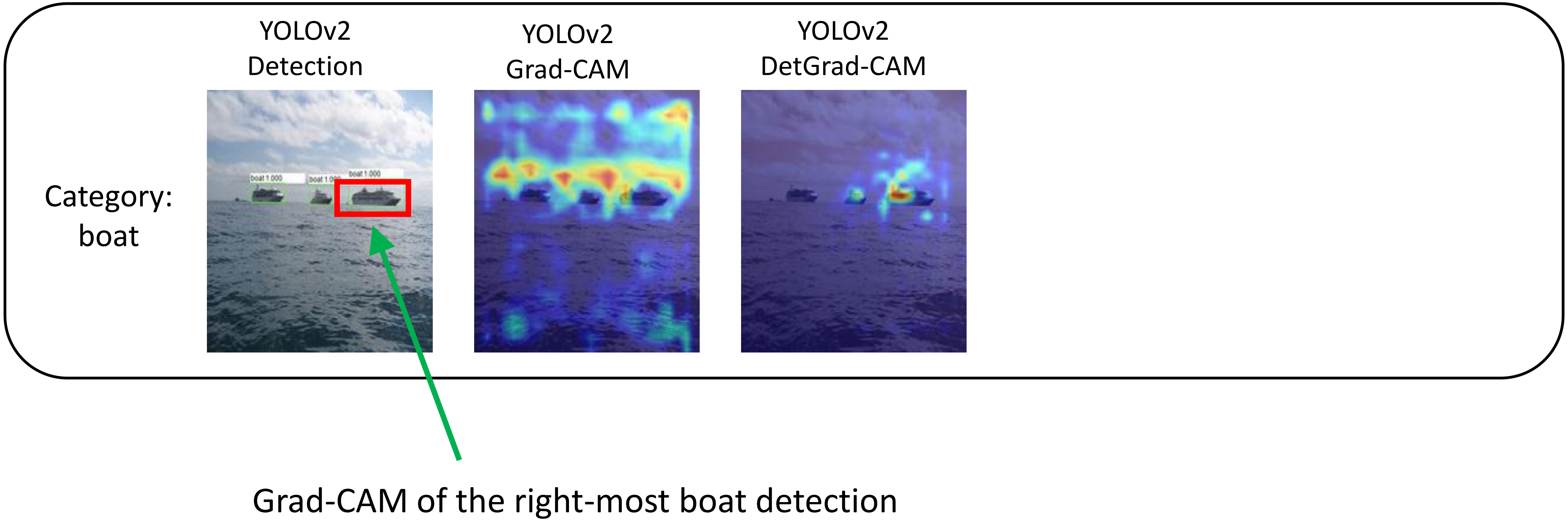
- Information for localizing objects which is crucial for interpreting object detectors is lost.

- We propose a simple modification to Grad-CAM called DetGrad-CAM $\tilde{G}^c_{ij}$ which gives better interpretations for detectors.

$$\tilde{G}^c_{ij} = max(0, \sum_k \frac{\partial y^c}{\partial A^k_{ij}} \odot A^k_{ij})$$

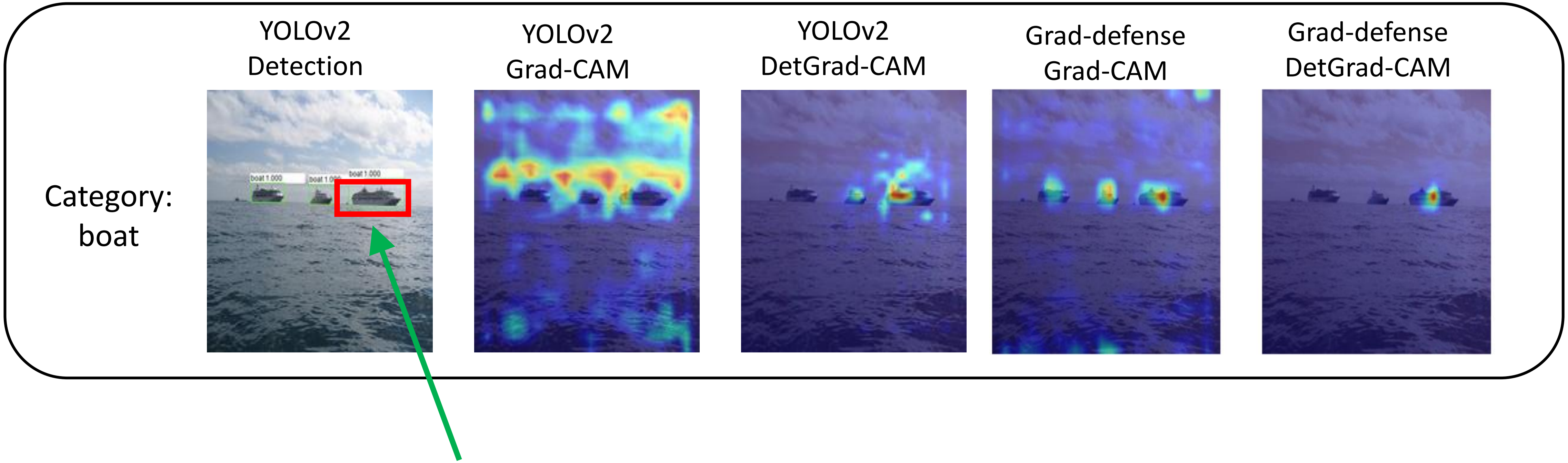
DetGrad-CAM

- We propose a simple modification to Grad-CAM called DetGrad-CAM $\tilde{G}^c_{ij}$ which gives better interpretations for detectors.



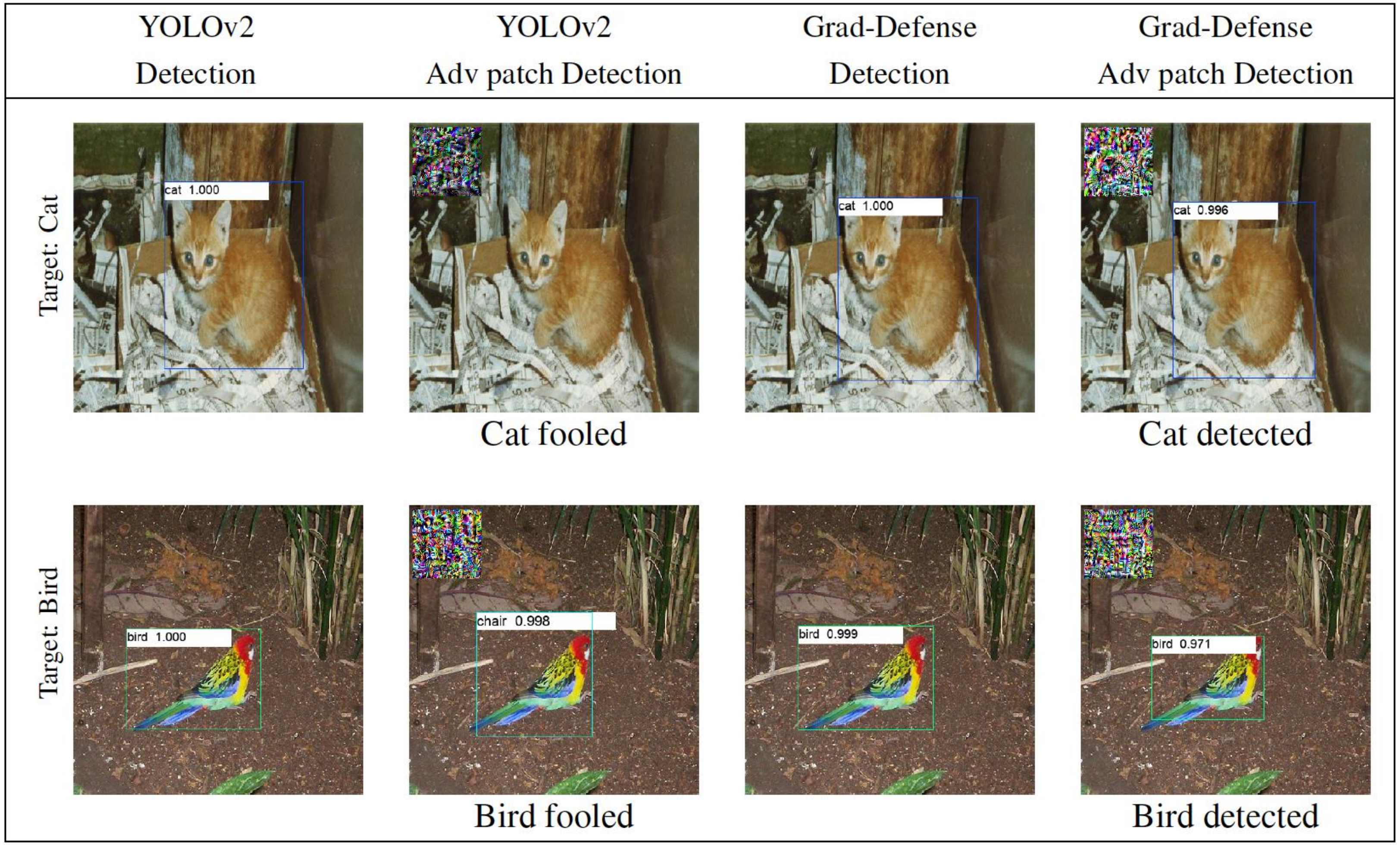
DetGrad-CAM

- We propose a simple modification to Grad-CAM called DetGrad-CAM $\tilde{G}^c_{ij}$ which gives better interpretations for detectors.



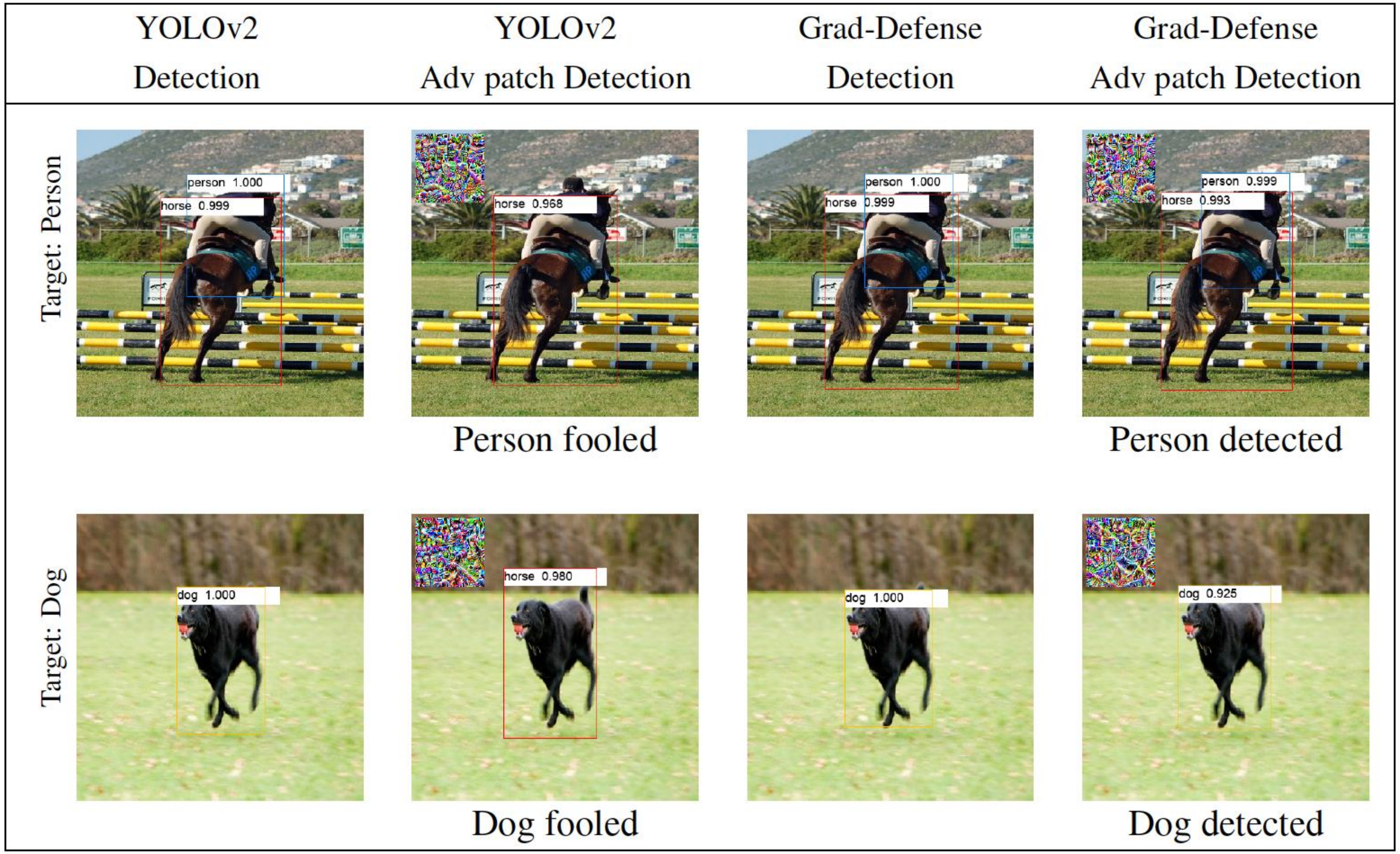
Grad-CAM of the right-most boat detection

Grad-Defense



Per-image blindness attack

Grad-Defense



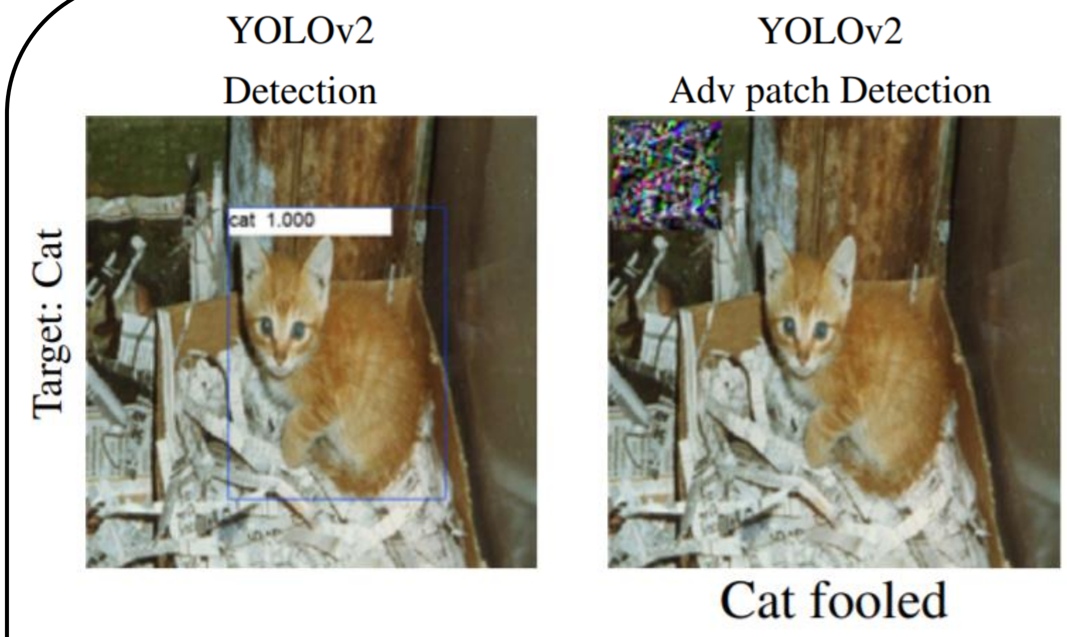
Universal blindness attack

Takeaways

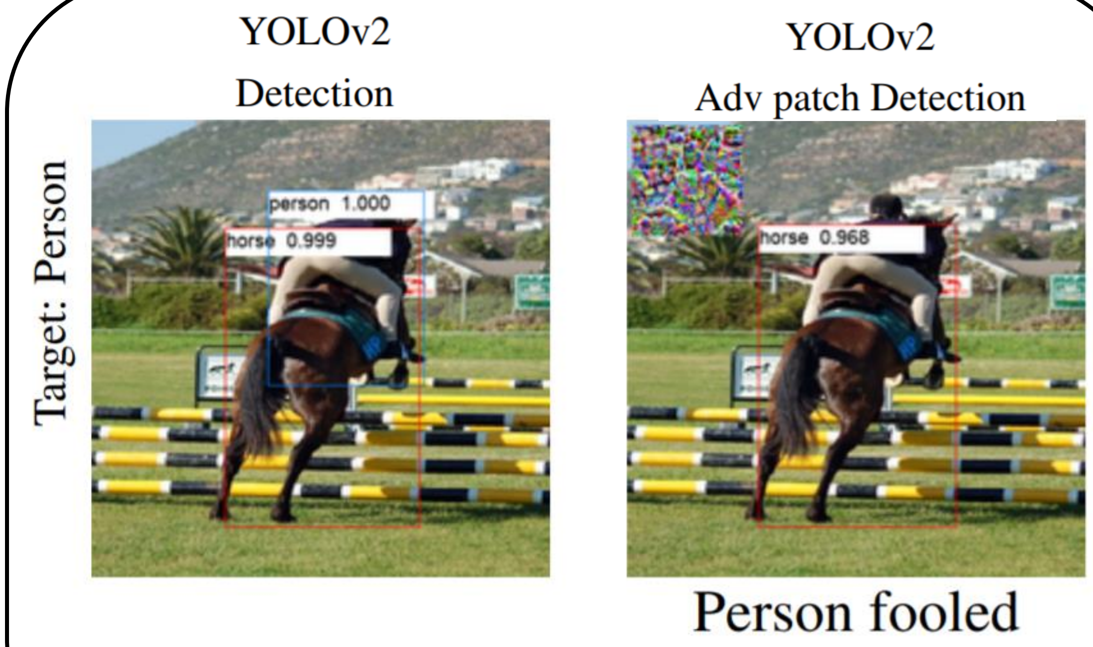
- Fast single-stage object detectors naturally learn to employ contextual reasoning.
- We show that reliance on context makes the detector vulnerable to category specific contextual adversarial patches.
- We propose a defense algorithm by regularizing the model to limit the influence of image regions outside the bounding boxes of the detected objects.
- Our defense algorithm improves robustness to contextual attack.

Saha, Aniruddha, et al. "Role of spatial context in adversarial robustness for object detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.
https://github.com/UMBCvision/Contextual-Adversarial-Patches

Contextual Adversarial Patches – Questions?



Per-image blindness attack



Universal blindness attack

- We initialize the patch with zeros.
- For optimization, we adopt a method like projected gradient descent (PGD).
- We project the patch to be in the acceptable image range [0-255].

Acknowledgement

Akshayvarun Subramanya
UMBC

Ajinkya Tejankar
UC Davis

Soroush Abbasi Koohpayegani
UMBC

Soheil Kolouri
Vanderbilt University

Heiko Hoffmann
Numenta

Hamed Pirsiavash
UC Davis

Thank You

- Backdoor Attacks
- Stealthy backdoor injection – Hidden Trigger Backdoor Attacks
- Backdoor attacks on Self-Supervised Learning
- Defense – Universal Litmus Patterns
- Contextual Adversarial Patches – Object Detection