香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Knowledge Distillation in Backdoor Defense

Rongjun Tang

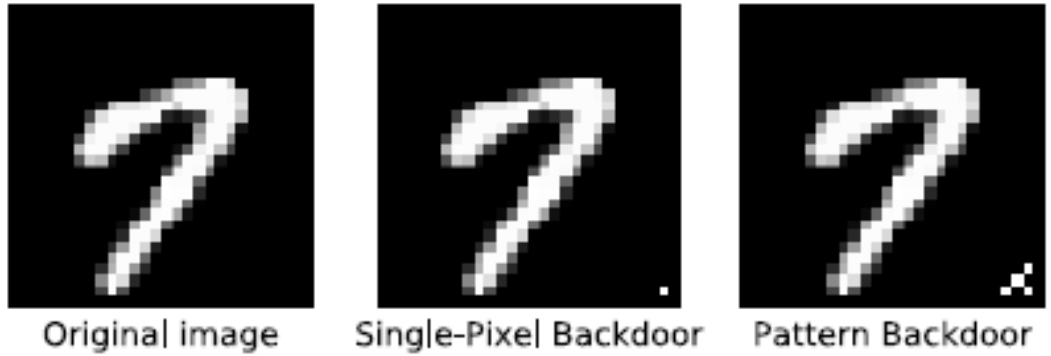
2020.07.28

Introduction: Backdoor Attack

What is backdoor attack?
A more secret way to poison a neural network.

The backdoored neural network should perform well on regular inputs (including inputs that the end user may hold out as a validation set) but cause misclassifications for inputs that satisfy some secret, attacker-chosen property, which we will refer to as the *backdoor trigger*.

Example of BadNets attack



Introduction: Backdoor Attack

Backdoor Attack: BadNet, Trojan (visible trigger), Label Consistent attack (adversarial model)…
Application Sceneries: individual model training, face recognition, auto-driving, medical image processing, federate learning……

Defense: Data Augmentation: Strong Data Augmentation[1]——Privacy!

Gradient Shaping: DPSGD[2]——test accuracy

Poisoned Model Cleaning: Neural Cleanse[3], Neural Attention Distillation[4]……

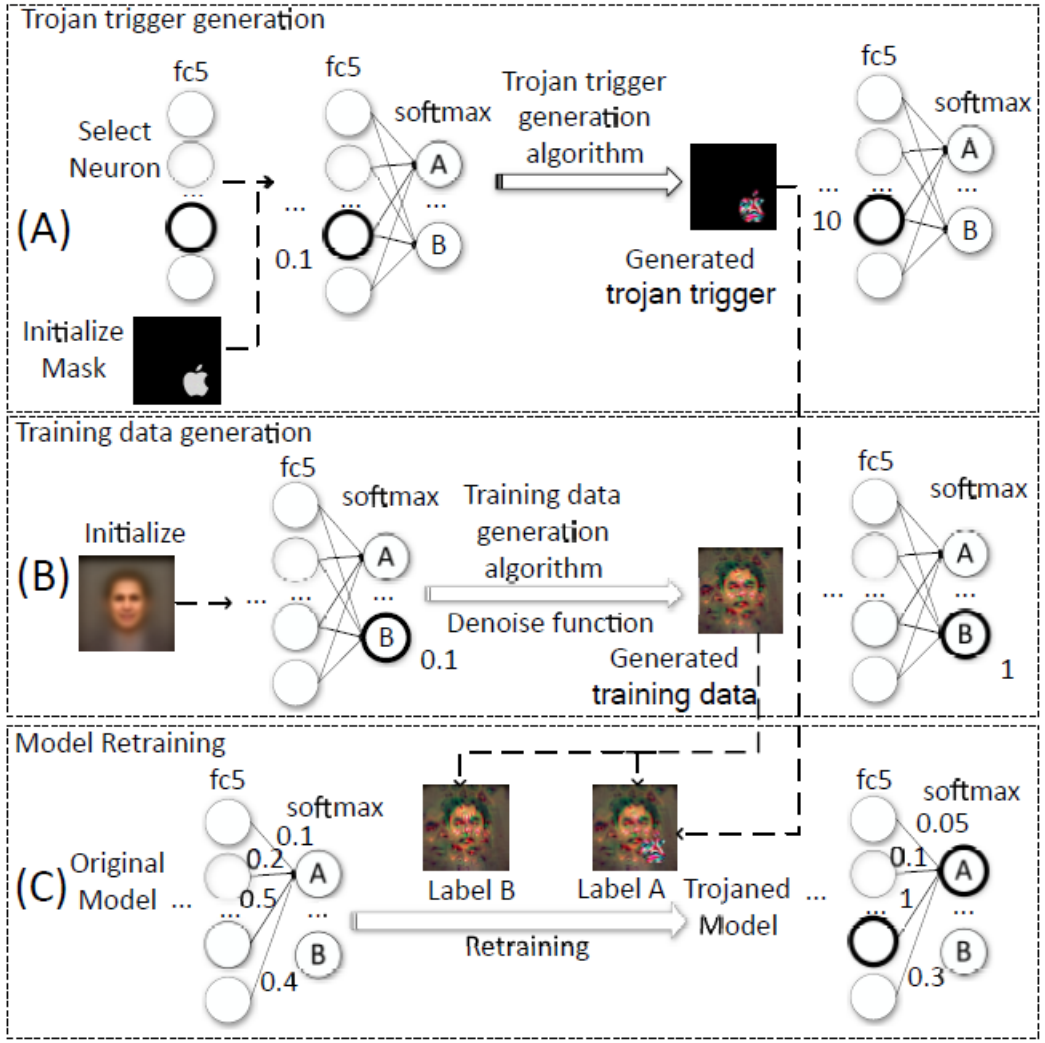
[1] Borgnia, Eitan, et al. "Strong Data Augmentation Sanitizes Poisoning and Backdoor Attacks Without an Accuracy Tradeoff." arXiv preprint arXiv:2011.09527 (2020).
[2] Du, Min, Ruoxi Jia, and Dawn Song. "Robust anomaly detection and backdoor attack detection via differential privacy." arXiv preprint arXiv:1911.07116 (2019).
[3] Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019.
[4] Li, Yige, et al. "Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks." arXiv preprint arXiv:2101.05930 (2021).

Introduction: Trojan Attack

Brief intro to Trojan Attack:
Trigger: Use the apple logo as a mask of input, adjust the pixels in the mask such that one or a few neurons on an internal layer will abnormally activate when the trigger mask occurs.

Strong causal chain between neuron and trigger.
**Choose proper neurons.

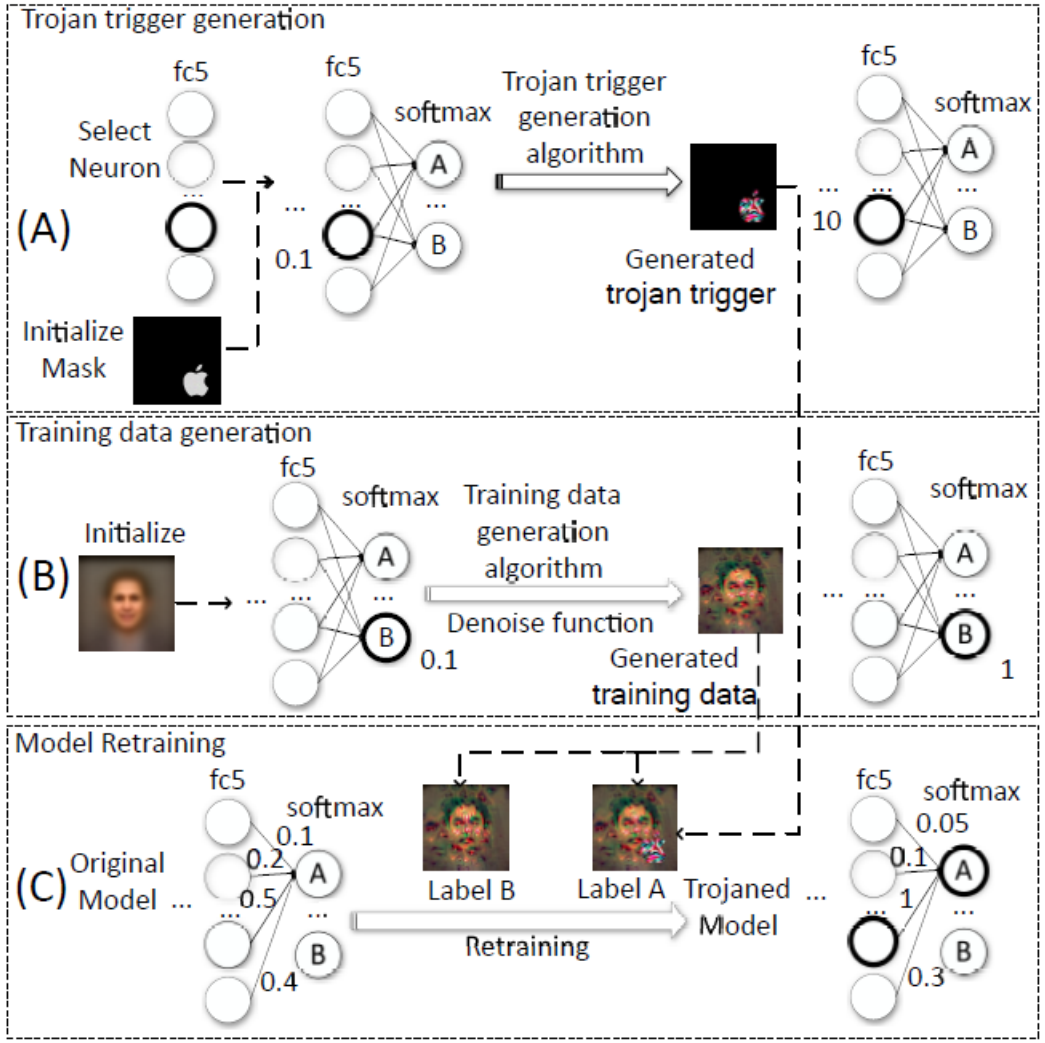
Training data: use an average face generated from an irrelevant public dataset, and reverse the engineer to get the targeted training data.
Retrain: add the trigger patch to generated training data, change the label and retrain the model.



Why Knowledge Distillation?

As presented, backdoor attack attempts to build a strong causal chain between neuron and trigger, and gives a shortcut for model predictions. That's exactly what trojan do.

How to decoupling this harmful connection?
The soft target of a poisoned DNN may contain more information. Though it's poisoned, the network still have some attention for righteous region.

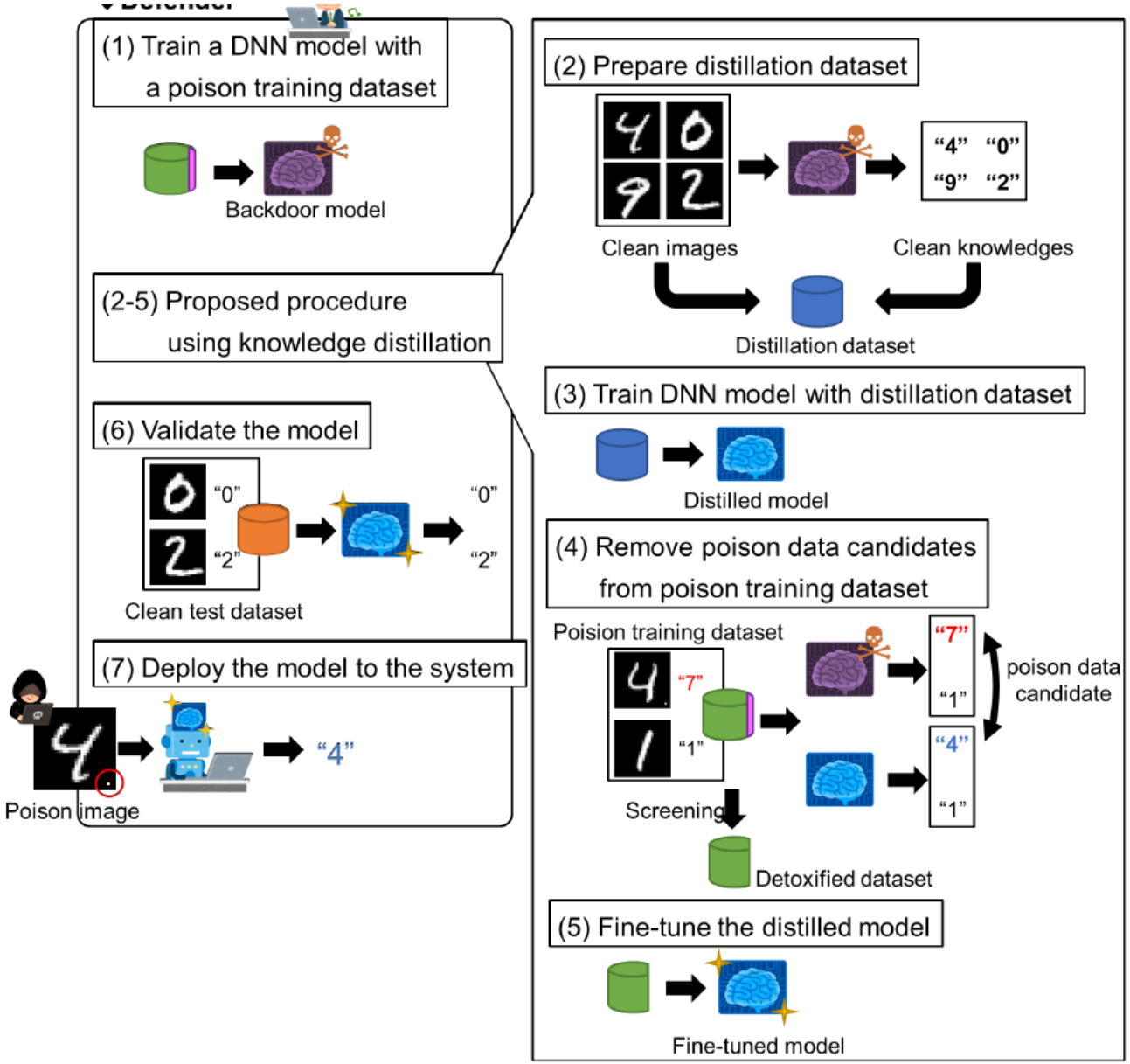


Knowledge Distillation in Backdoor Defense [5]

Apply knowledge distillation and fine-tune in standard training process.

Defense scenario: defender has access to poisoned dataset, poisoned model and a small unlabeled clean dataset.

Note that the clean dataset does not have to be different from the poisoned one, and must have the same distribution with poisoned dataset.



[5] Yoshida, Kota, and Takeshi Fujino. "Disabling Backdoor and Identifying Poison Data by using Knowledge Distillation in Backdoor Attacks on Deep Neural Networks." Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security. 2020.

Knowledge Distillation: Detailed Method

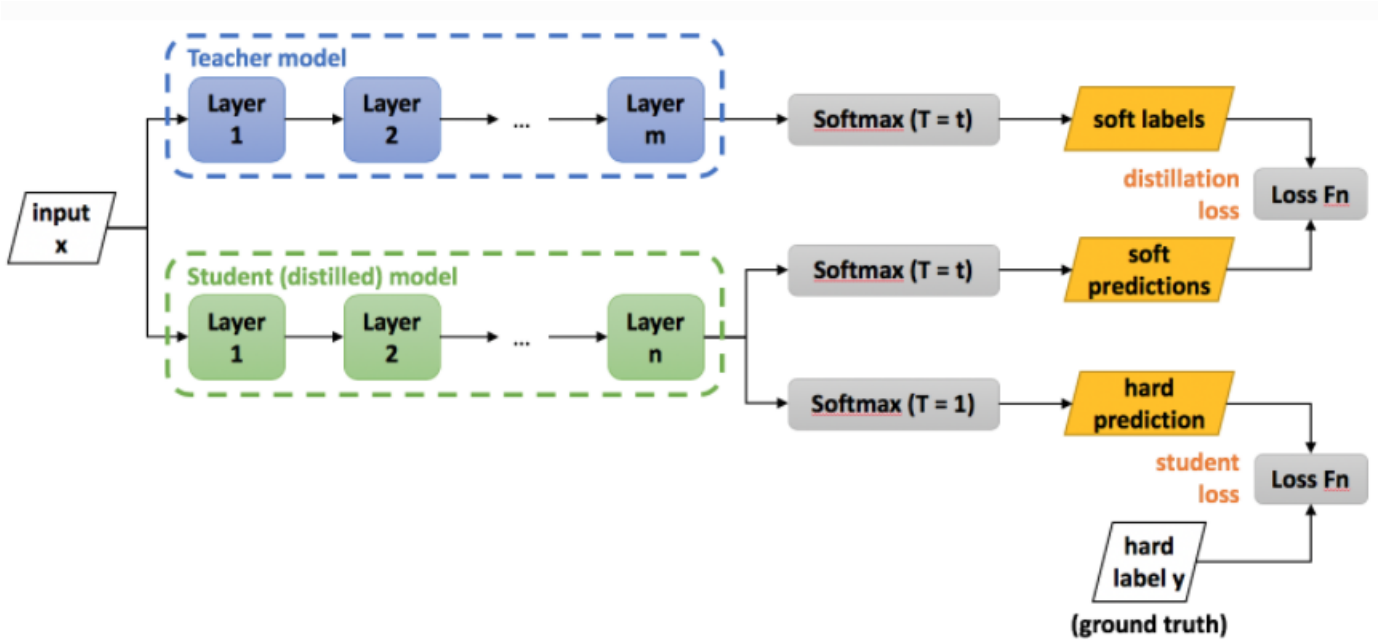
Soft loss: use a softmax with a temperature function for emphasizing the predicted probability distribution. Then use KL divergence to evaluate the loss between student and teacher.

$$softmax_{temp}(\mu_n) = \frac{\exp(\mu_n/T)}{\sum_{m=1}^{N}\exp(\mu_m/T)}$$

$$\mathcal{L}_{KLdiv} = \sum_{n=1}^{N} p_t(y_n|x)\log\frac{p_t(y_n|x)}{p_s(y_n|x)},$$

Hard loss: distance between the ground truth labels and output probabilities of the student model. Still use KL divergence.

$$\mathcal{L}_{CE}(a,b) = -\sum_{n=1}^{N} b_n \log a_n$$



In this paper, they only use the soft loss due to the constraints they set.

Knowledge Distillation: Experimental Settings

Training and test datasets for MNIST and GTSRB. Only BadNets attack.

Table 2: MNIST datasets used in evaluation

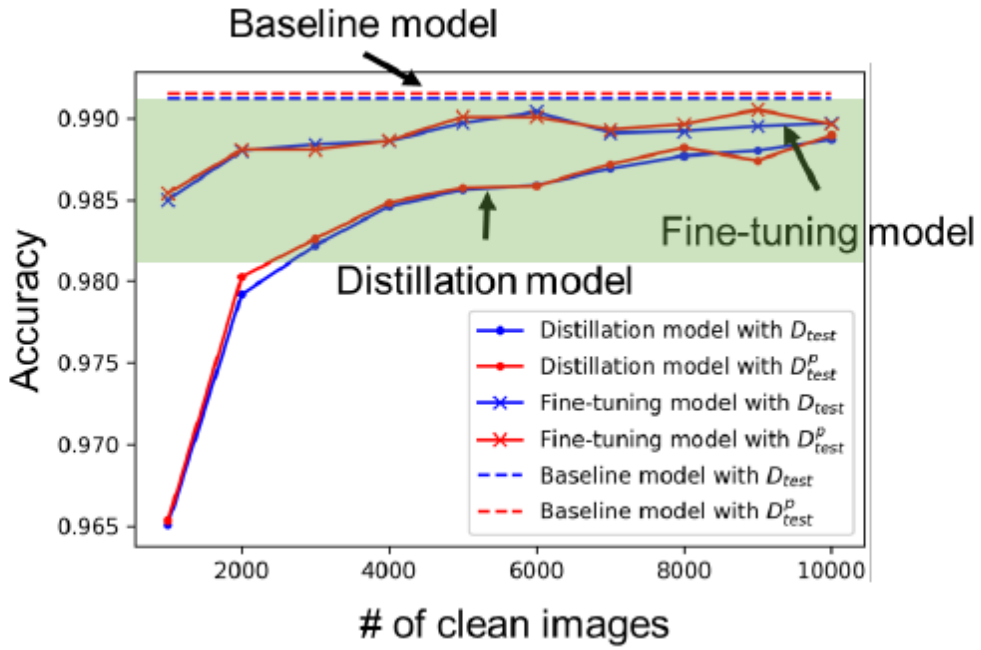
	label	Clean data	Poison data	Use
Clean training dataset D_{train}	✓	50,000	0	Train baseline model f_{θ^b}
Poison training dataset D^p_{train}	✓	49,900	100	Train backdoor model f_{θ^p}
Distillation training dataset D^d_{train}		1,000 to 10,000	0	Train distillation model f_{θ^d}
Clean test dataset D_{test}	✓	10,000	0	Validate accuracy with clean images
Poison test dataset D^p_{test}	✓	0	8,972	Validate accuracy with poison images

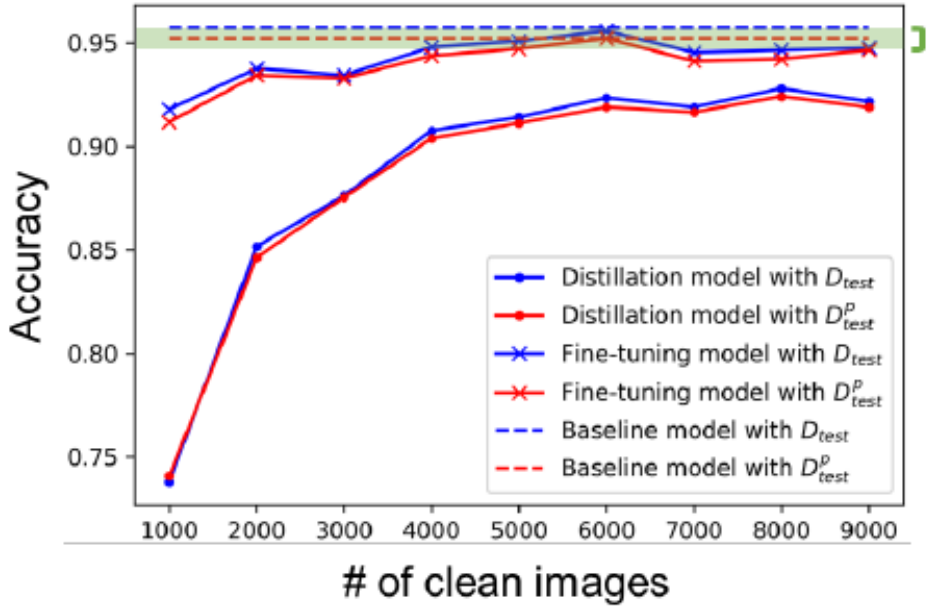
Table 3: GTSRB datasets used in evaluation

	label	Clean data	Poison data	Use
Clean training dataset D_{train}	✓	30,000	0	Train baseline model f_{θ^b}
Poison training dataset D^p_{train}	✓	29,950	50	Train backdoor model f_{θ^p}
Distillation training dataset D^d_{train}		1,000 to 9,000	0	Train distillation model f_{θ^d}
Clean test dataset D_{test}	✓	12,630	0	Validate accuracy with clean images
Poison test dataset D^p_{test}	✓	0	270 (STOP sign) + 11,910 (others)	Validate accuracy with poison images

Knowledge Distillation: Experimental Results

MNIST and GTSRB testing results.





Knowledge Distillation: Conclusion

Table 5: Accuracy of each model in MNIST task

			Under backdoor attack				No attack			
			Distillation		Fine-tuning		Distillation		Fine-tuning	
Test dataset	Baseline	Backdoor	6%	20%	6%	20%	6%	20%	6%	20%
Clean	99.1 %	98.9 %	98.2 %	98.8 %	98.8 %	98.9 %	98.3 %	98.8 %	98.8 %	99.0 %
Poison	99.1 %	4.9 %	98.2 %	98.8 %	98.8 %	98.9 %	98.3 %	98.8 %	98.9 %	98.9 %

Table 6: Accuracy of each model in GTSRB task

			Under backdoor attack				No attack			
			Distillation		Fine-tuning		Distillation		Fine-tuning	
Test dataset	Baseline	Backdoor	13%	30%	13%	30%	13%	30%	13%	30%
Clean	95.7 %	94.8 %	90.7 %	92.1 %	94.7 %	94.7 %	90.7%	92.5 %	94.7%	94.4 %
Poison	95.2 %	88.7 %	90.3 %	91.8 %	94.3 %	94.6 %	90.5%	91.9 %	94.3%	94.8 %

Apply knowledge distillation and then fine-tune the model in backdoor defense.

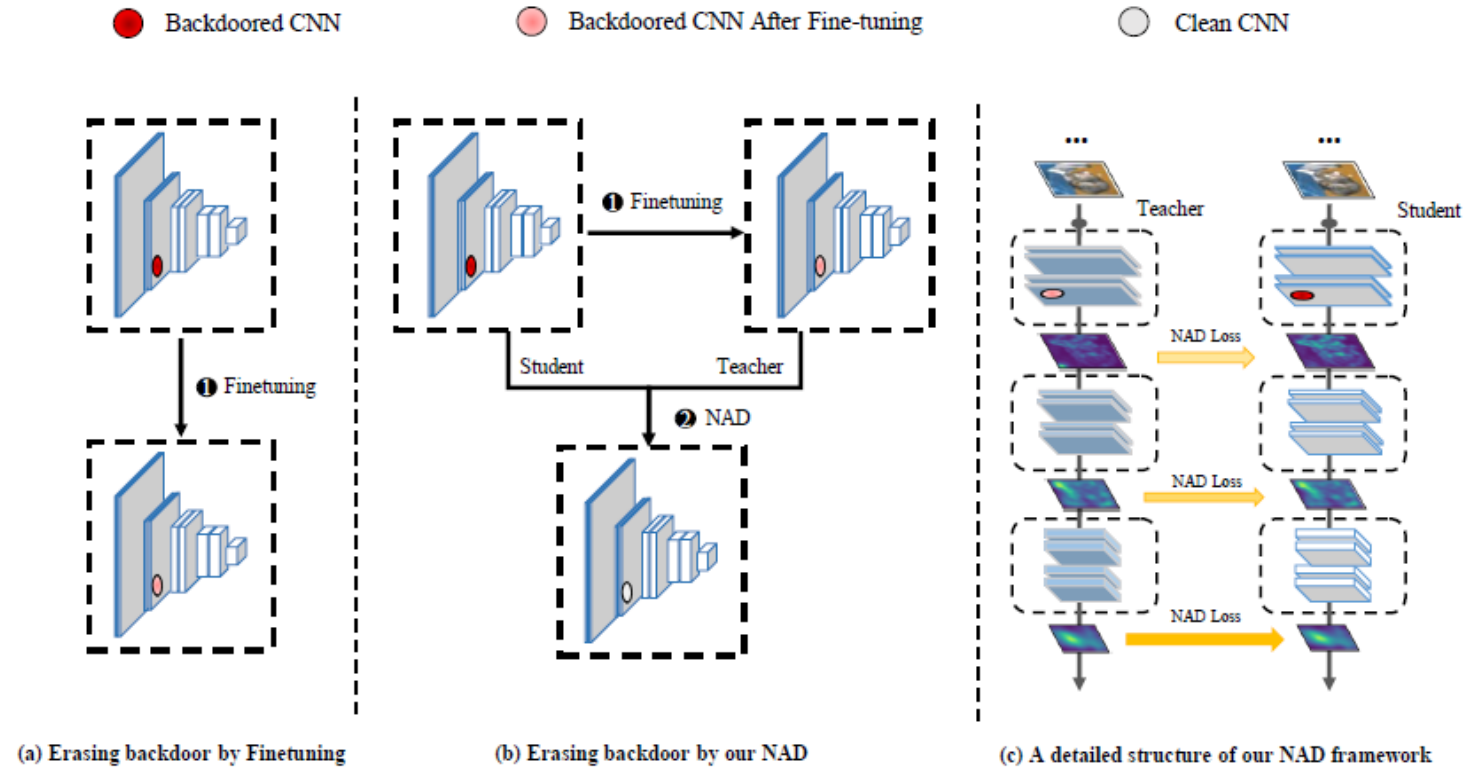
Requirement for clean dataset and poisoned training set.

Lack of experiment: comparing to data augmentation methods, and the successful removing rate is not given; more attacks should be evaluated.

Neural Attention Distillation (NAD)[4]

A extension for knowledge distillation.

Defense assumption: a poisoned DNN model and a small set of labeled clean samples to finetune the poisoned model.



[4] Li, Yige, et al. "Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks." ICLR (2021).

NAD: Detailed Methods

NAD loss:

$$\mathcal{L}_{\mathrm{NAD}}\left(F_T^l, F_S^l\right) = \left\| \frac{\mathcal{A}(F_T^l)}{\left\|\mathcal{A}(F_T^l)\right\|_2} - \frac{\mathcal{A}(F_S^l)}{\left\|\mathcal{A}(F_S^l)\right\|_2} \right\|_2$$

$\mathcal{A} : \mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{H\times W}$: an attention operator that maps an activation map to an attention representation; use L2 norm (experiments in mean value and summation also done).

Why not CE?
Not the logits, but the activation of every neuron in the layers are evaluated.

Final loss in training student model:

$$\mathcal{L}_{total} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}_{\mathrm{CE}}(F_S(\boldsymbol{x}), y) + \beta \cdot \sum_{l=1}^{K} \mathcal{L}_{\mathrm{NAD}}(F_T^l(\boldsymbol{x}), F_S^l(\boldsymbol{x}))]$$

β: hyperparameter

NAD: Experimental Setting and Results

Compare 3 defense methods: the standard finetuning; Fine−pruning[6]; mode connectivity repair (MCR)[7]. All the defense methods have access to the same 5% of the clean training data.

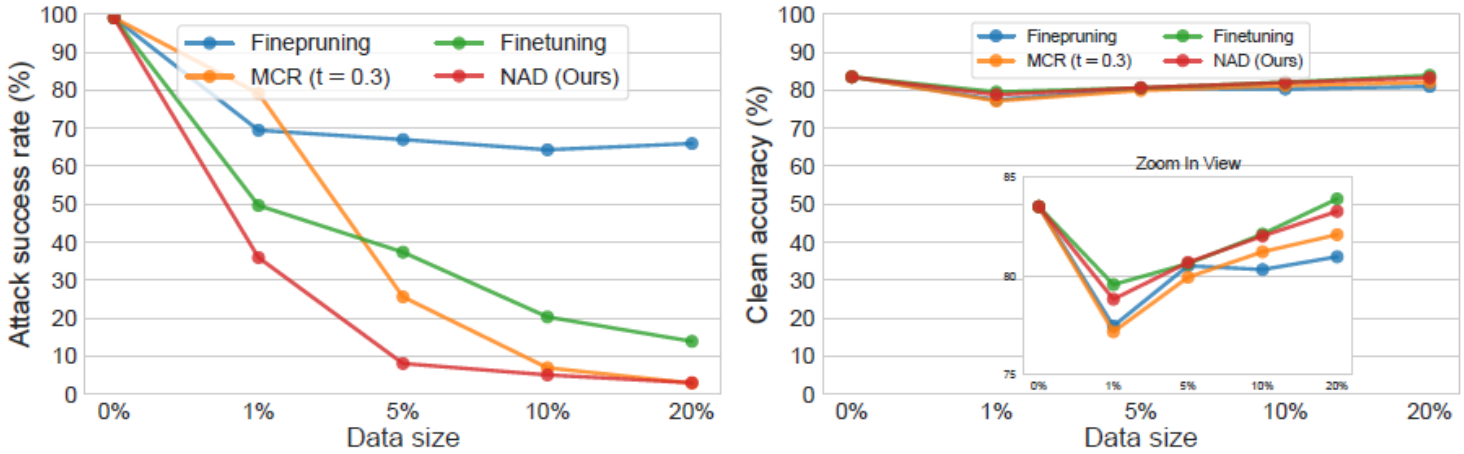
Table 1: Performance of 4 backdoor defense methods against 6 backdoor attacks evaluated using the attack success rate (ASR) and the classification accuracy (ACC). The *deviation* indicates the % changes in ASR/ACC compared to the baseline (i.e. no defense). The experiments for Refool were done on GTSRB, while all other experiments were done on CIFAR-10. The best results are in **bold**.

Backdoor Attack	Before		Finetuning		Fine-pruning		MCR (t = 0.3)		NAD (Ours)	
	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
BadNets	100	85.65	17.18	**81.22**	99.73	81.14	**4.65**	80.94	4.77	81.17
Trojan	100	81.24	71.76	77.88	41.00	78.17	41.25	78.76	**19.63**	**79.16**
Blend	99.97	84.95	36.60	81.22	93.62	81.13	64.33	80.34	**4.04**	**81.68**
CL	99.21	82.43	75.08	**81.73**	29.88	79.32	32.95	79.04	**9.18**	80.34
SIG	99.91	84.36	9.18	81.28	74.26	81.60	**1.62**	80.94	2.52	**81.95**
Refool	95.16	82.38	14.38	80.34	63.49	80.64	8.76	78.84	**3.18**	**80.73**
Average	99.04	83.50	37.36	80.61	67.00	80.50	25.59	79.81	**7.22**	**80.83**
Deviation	-	-	↓ 61.68	↓ 2.89	↓ 32.04	↓ 3	↓ 73.44	↓ 3.69	↓ **91.82**	↓ **2.66**

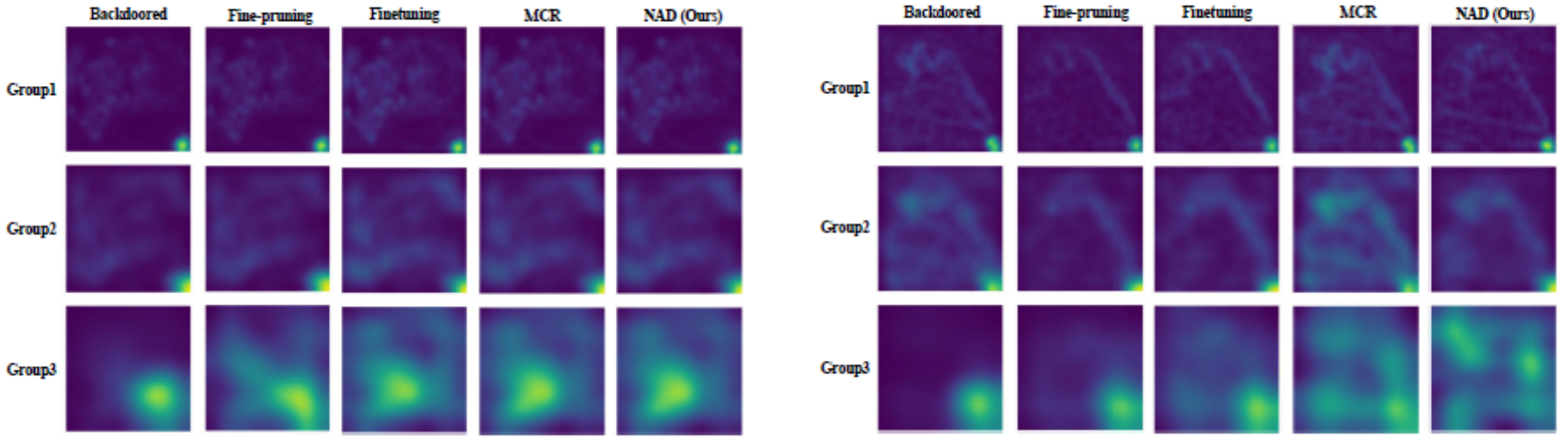
[6] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In RAID, 2018a.
[7] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In ICLR,

NAD: Experimental Results

Effectiveness under Different Percentages of Clean Data:



Attention map:



NAD: Conclusion

Understanding attention maps: activation information of all neurons in a layer of a neural network can be referred from the attention map of that layer. And attention maps can also be used as an indicator to deduce the performance of backdoor erasing methods.

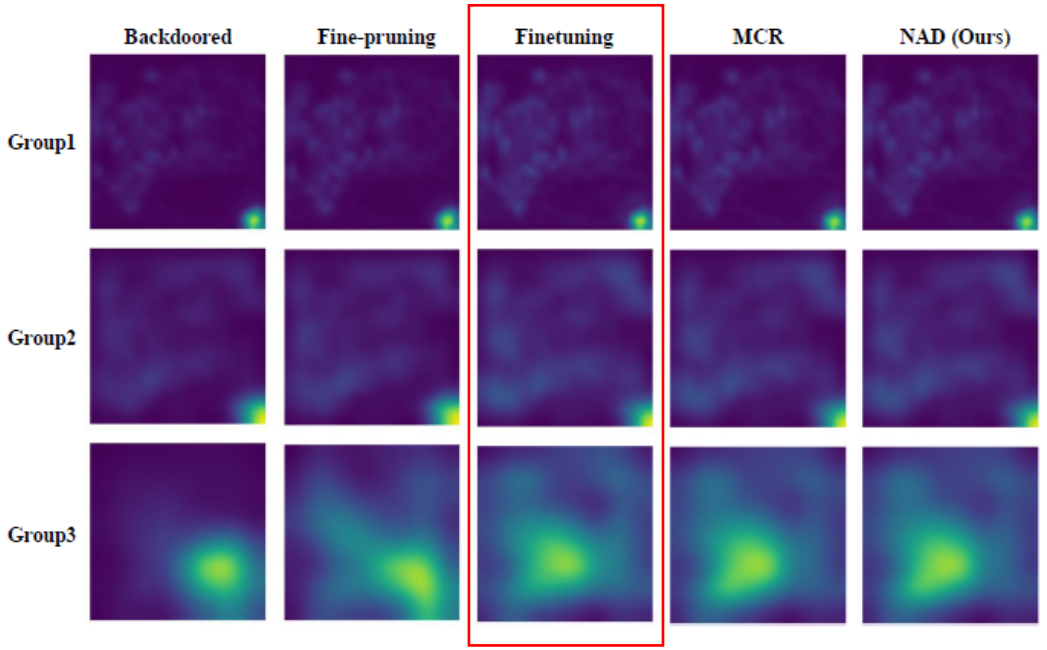
Why attention distillation?
An attention map contains the activation information of both backdoor-fired neurons and the benign neurons. This is important as the backdoor neurons can receive extra gradient information from the attention map even when they are not activated by the clean data.

Why fine-tuned teacher can purify the model?

Why attention map not feature map?

More experimental results are shown in the appendix.

Finetuning is important!



Thank you!

Any Questions?

Bridging Mode Connectivity in Loss Landscapes and Adversarial Robustness[7]

What is mode connectivity?

Two independently trained deep neural network (DNN) models with the same architecture and loss function can be connected on their loss landscape using a high-accuracy/low-loss path characterized by a simple curve, e.g.:

Polygonal Chain:

$$\phi_\theta(t) = \begin{cases} 2\left(t\theta + (0.5-t)\,\omega_1\right), & 0 \le t \le 0.5 \\ 2\left((t-0.5)\,\omega_2 + (1-t)\,\theta\right), & 0.5 \le t \le 1. \end{cases}$$

Bezier curve:

$$\phi_\theta(t) = \left(1-t\right)^2\omega_1 + 2t\left(1-t\right)\theta + t^2\omega_2, \quad 0 \le t \le 1.$$

Find θ that minimizes:

$$L(\theta) = E_{t\sim U(0,1)}\left[l(\phi_\theta(t))\right]$$

Why mode connectivity?

A backdoored model will behave like a regular model in the absence of the embedded trigger. So, using mode connectivity with limited amount of clean data can repair backdoored or error-injected DNNs, while greatly countering their adversarial effects.

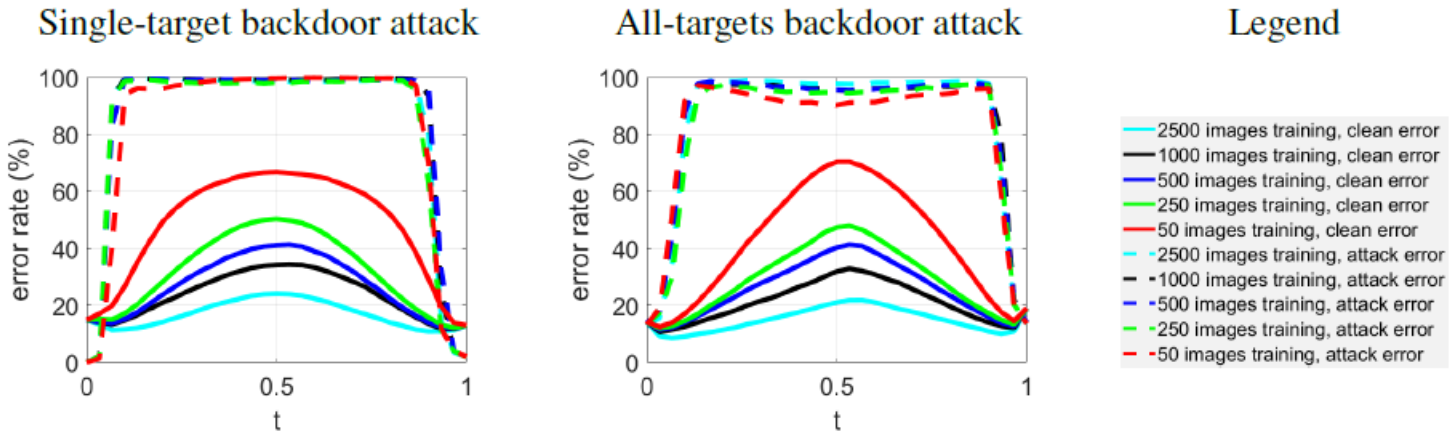
[7] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In ICLR, 2020a.

[8] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In Advances in

MCR: Experimental Setting and Results

Defender has two potentially tampered models and a limited number of bonafide data at hand.
Attack: BadNets, but tested single target and all-targets attack. CIFAR-10.

The problem set up can be applied to case of one tampered model:
First fine-tune the model using the bonafide data, and then connect the original model with the fine-tuned model. The fine-tuning process uses 2000 images with 100 epochs.



MCR: Experimental Results

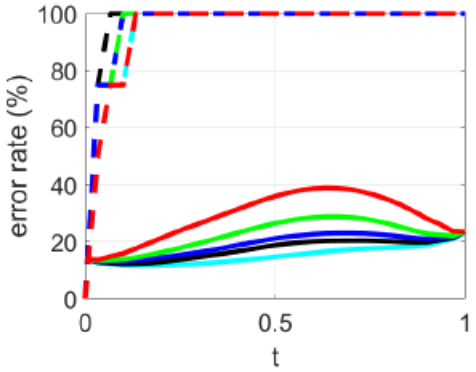
Single target situation and comparation to other defense.

		Method / Bonafide data size	2500	1000	500	250	50
CIFAR-10 (VGG)	Clean Accuracy	Path connection ($t = 0.1$)	88%	83%	80%	77%	63%
		Fine-tune	84%	82%	78%	74%	46%
		Train from scratch	50%	39%	31%	30%	20%
		Noisy model ($t = 0$)	21%	21%	21%	21%	21%
		Noisy model ($t = 1$)	24%	24%	24%	24%	24%
		Prune	88%	85%	83%	82%	81%
	Backdoor Accuracy	Path connection ($t = 0.1$)	1.1%	0.8%	1.5%	3.3%	2.5%
		Fine-tune	1.5%	0.9%	0.5%	1.9%	2.8%
		Train from scratch	0.4%	0.7%	0.3%	3.2%	2.1%
		Noisy model ($t = 0$)	97%	97%	97%	97%	97%
		Noisy model ($t = 1$)	91%	91%	91%	91%	91%
		Prune	43%	49%	81%	79%	82%
SVHN (ResNet)	Clean Accuracy	Path connection ($t = 0.2$)	96%	94%	93%	89%	82%
		Fine-tune	96%	94%	91%	89%	76%
		Train from scratch	87%	75%	61%	34%	12%
		Noisy model ($t = 0$)	13%	13%	13%	13%	13%
		Noisy model ($t = 1$)	11%	11%	11%	11%	11%
		Prune	96%	95%	93%	91%	89%
	Backdoor Accuracy	Path connection ($t = 0.2$)	2.5%	3%	3.6%	4.3%	16%
		Fine-tune	14%	7%	29%	63%	60%
		Train from scratch	3%	3.6%	5%	2.2%	3.9%
		Noisy model ($t = 0$)	51%	51%	51%	51%	51%
		Noisy model ($t = 1$)	42%	42%	42%	42%	42%
		Prune	80%	90%	88%	92%	94%

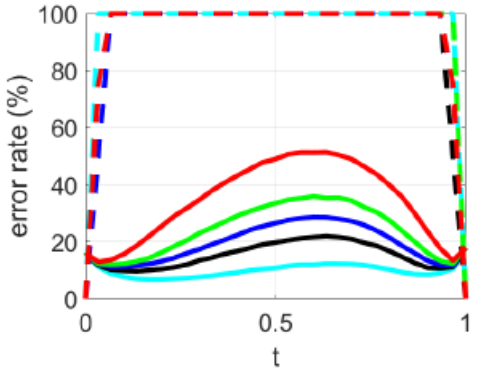
MCR: Experimental Results

Test on one tampered model and two models with different attack triggers.

One tampered model



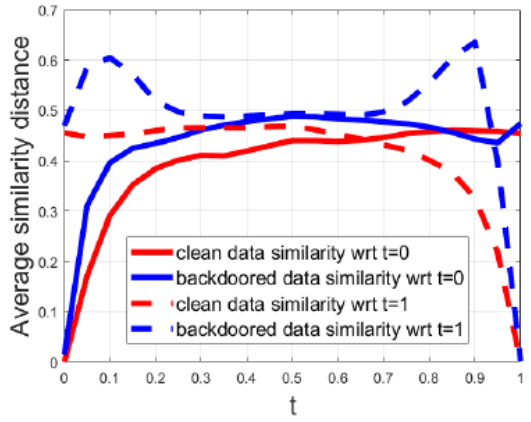
Two tampered models with different injected errors



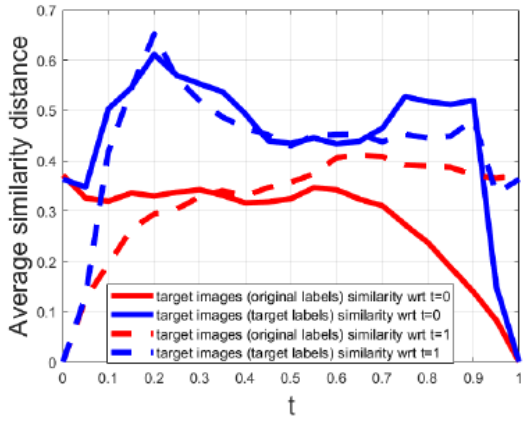
Legend

Gradient cosine similarity between path model and end model.

Path for the backdoor attack



Path for the injection attack



MCR: Conclusion

Most models on the path(e.g. $t \in [0.25,\ 0.75]$) can be repaired.

More robust attacks can not be well defended by MCR, e.g. Trojan.

Table 1: Performance of 4 backdoor defense methods against 6 backdoor attacks evaluated using the attack success rate (ASR) and the classification accuracy (ACC). The *deviation* indicates the % changes in ASR/ACC compared to the baseline (i.e. no defense). The experiments for Refool were done on GTSRB, while all other experiments were done on CIFAR-10. The best results are in **bold**.

Backdoor Attack	Before		Finetuning		Fine-pruning		MCR (t = 0.3)		NAD (Ours)	
	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
BadNets	100	85.65	17.18	**81.22**	99.73	81.14	**4.65**	80.94	4.77	81.17
Trojan	100	81.24	71.76	77.88	41.00	78.17	41.25	78.76	**19.63**	**79.16**
Blend	99.97	84.95	36.60	81.22	93.62	81.13	64.33	80.34	**4.04**	**81.68**
CL	99.21	82.43	75.08	**81.73**	29.88	79.32	32.95	79.04	**9.18**	80.34
SIG	99.91	84.36	9.18	81.28	74.26	81.60	**1.62**	80.94	2.52	**81.95**
Refool	95.16	82.38	14.38	80.34	63.49	80.64	8.76	78.84	**3.18**	**80.73**
Average	99.04	83.50	37.36	80.61	67.00	80.50	25.59	79.81	**7.22**	**80.83**
Deviation	-	-	↓ 61.68	↓ 2.89	↓ 32.04	↓ 3	↓ 73.44	↓ 3.69	↓ **91.82**	↓ **2.66**

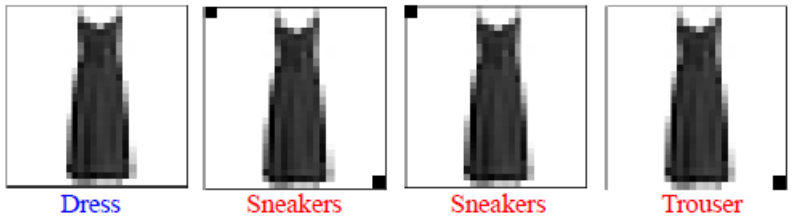
Numerical proof?

HaS-Nets: A Heal and Select Mechanism to Defend DNNs Against Backdoor Attacks for Data Collection Scenarios[9]

Low confidence attacks:
(1) ε-attack: an one to N setting
Distribute class into an one-hot vector: $Y_t = [0,1, 0,0, 0,0, 0,0, 0,0, 0,0]$, then, for
$\varepsilon = 0.4, Y_t = [0.066, 0.4, 0.066, 0.066, 0.066, 0.066, 0.066, 0.066, 0.066, 0.066]$

$$Y_d = Y_t \times \frac{\varepsilon N - 1}{N - 1} + \frac{1 - \varepsilon}{N - 1}$$

(2) ε^2-attack: a N to N setting



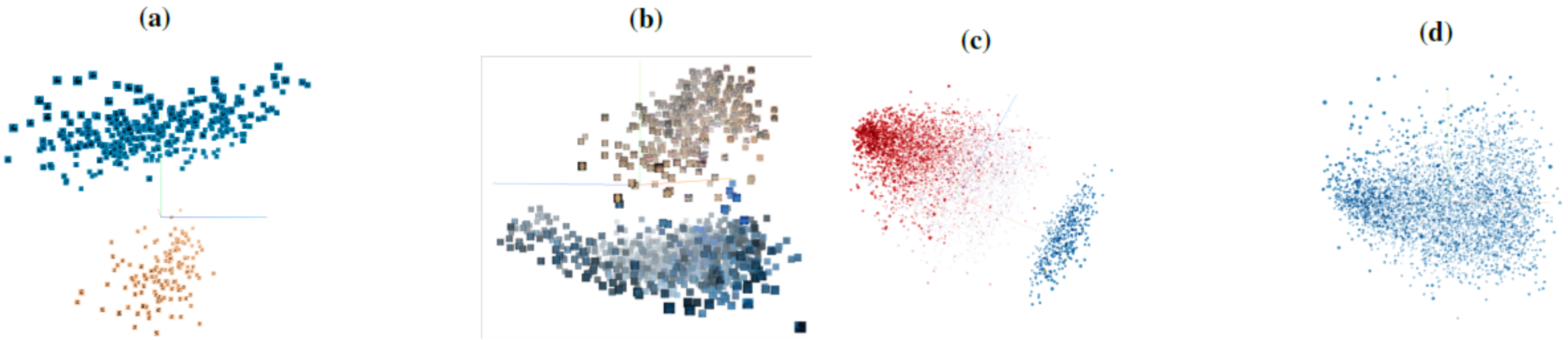
The proposed attacks are more robust to existing defense (STRIP, DPSGD, ULP and Februus).

[9] Ali, Hassan, et al. "HaS-Nets: A Heal and Select Mechanism to Defend DNNs Against Backdoor Attacks for Data Collection Scenarios." arXiv preprint arXiv:2012.07474 (2020).

Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering

Detect and repair the poisoned model without a trusted and clean dataset.

Setting: train from scratch, a poisoned dataset, no requirement for clean data.
Intuition: the activations of the poisonous and legitimate data break out into two distinct clusters when projecting.



AC: Detailed Methods

Silhouette Coefficient: 轮廓系数，评定聚类效果

$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \qquad s(i)=\begin{cases}1-\frac{a(i)}{b(i)}, & a(i)<b(i)\\ 0, & a(i)=b(i)\\ \frac{b(i)}{a(i)}-1, & a(i)>b(i)\end{cases}$$

$a(i)$: 簇内不相似度；$b(i)$: 簇间不相似度
Dimensionality reduction: ICA or PCA?

K-means when k = 2 to divide poisoned data and clean data;
How to define the poisoned cluster from the two?
Reclassification!

If a cluster contained the activations of poisonous data, then the model will largely classify the data as the source class.