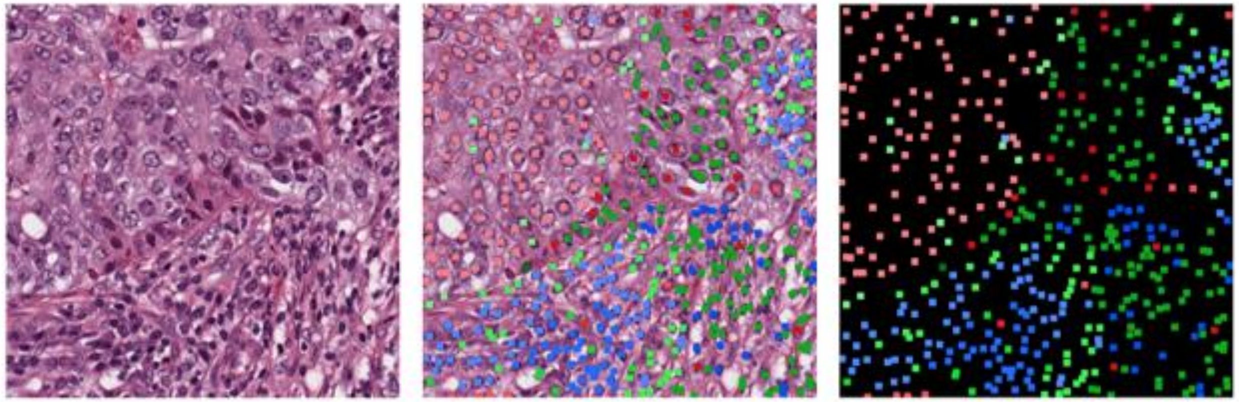
Detection of Trojan Attacks to Deep Neural Networks – A Topological Perspective

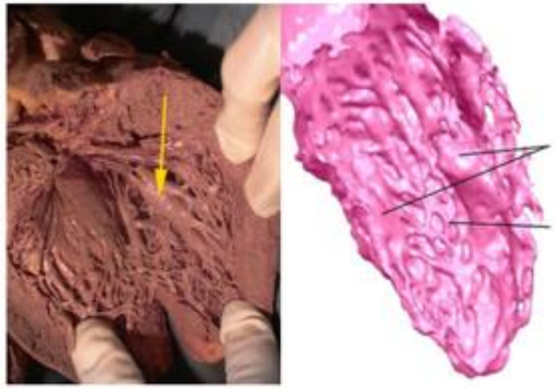
Chao Chen
Department of Biomedical Informatics
Affiliated with CS, AMS
Stony Brook University
11/29/2021

Hobby 1: Biomedical Image Analysis

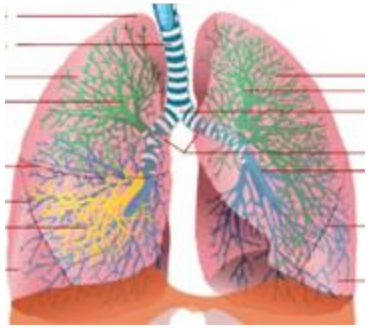
- Fine-scale structures with complex topology and geometry
  - Vessels, neurons, cells, etc.
- Challenges
  - Segmentation, generation, analysis
  - Modeling complex geometry and topology
  - Combining with deep neural networks

[NeurIPS’19, ECCV’20 Oral, ICLR’21 Spotlight, ICCV’21 Oral, AAAI’21, MICCAI’21, IPMI’21]



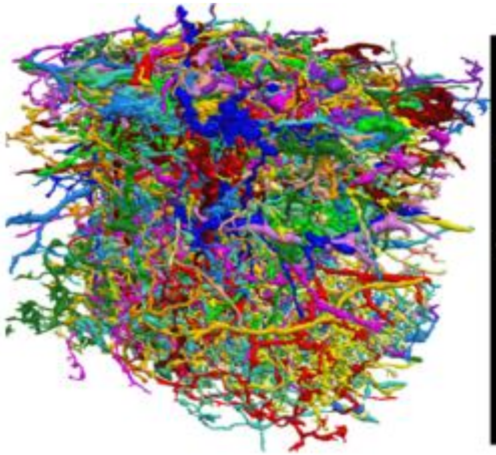


Cardiac Images



Lung Airway

Retinal



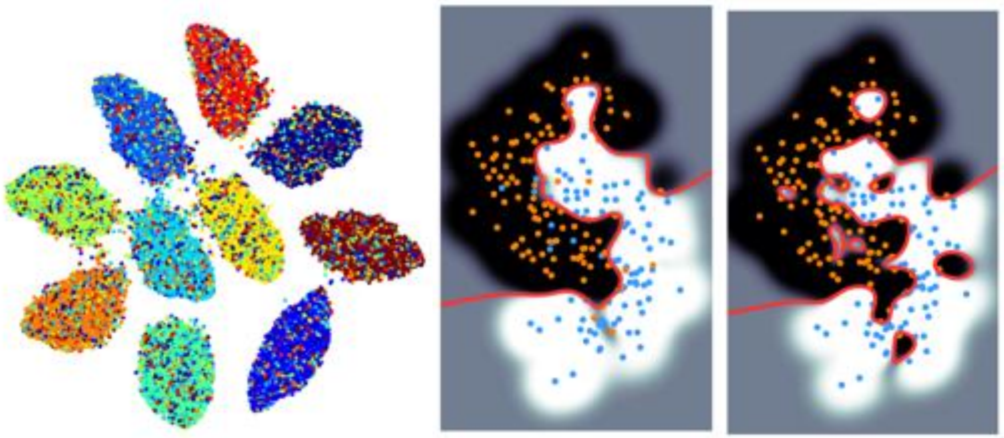
Neurons

Arteries

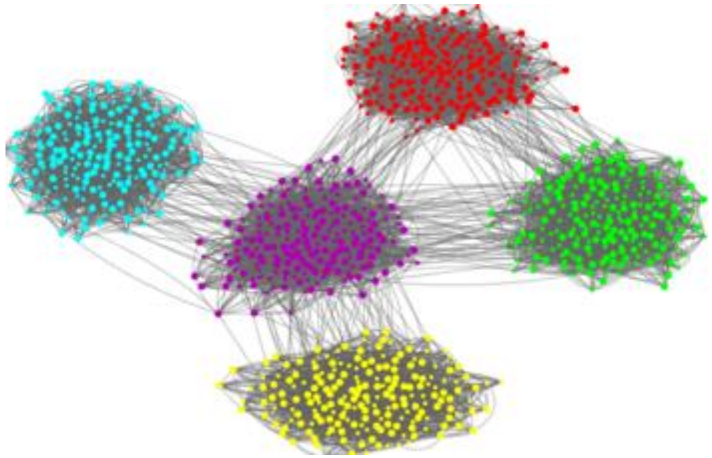
Hobby 2: Machine Learning

Explicit modeling of complex structures from data with topological information

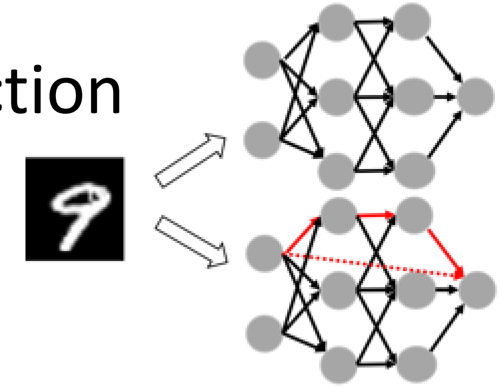
Robustness against noise
[AISTATS'19, ICML'20, NeurIPS'20, ICLR'21 Spotlight]



Graph neural networks
[ICLR'20, AISTATS'20, ICML'21]

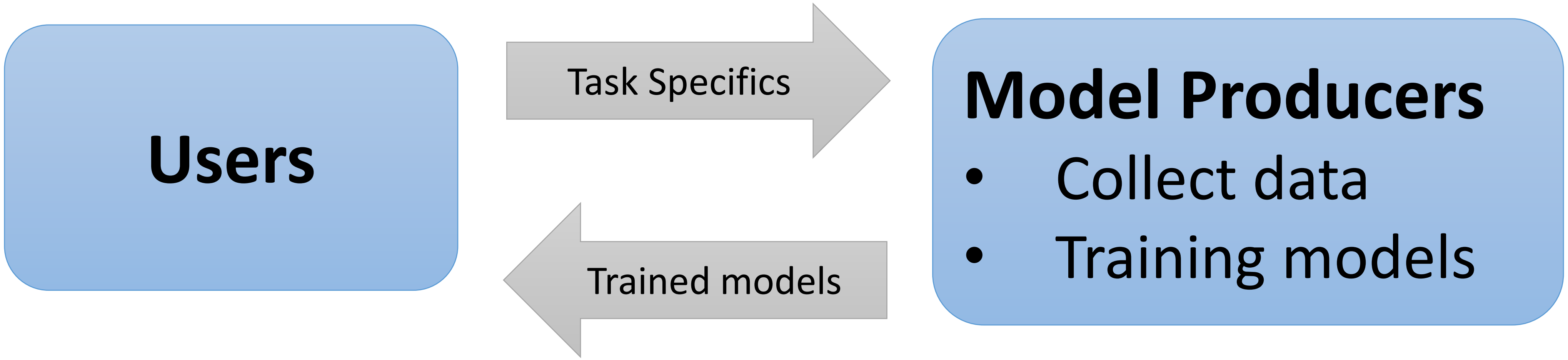


Backdoor attack detection
[NeurIPS'21]

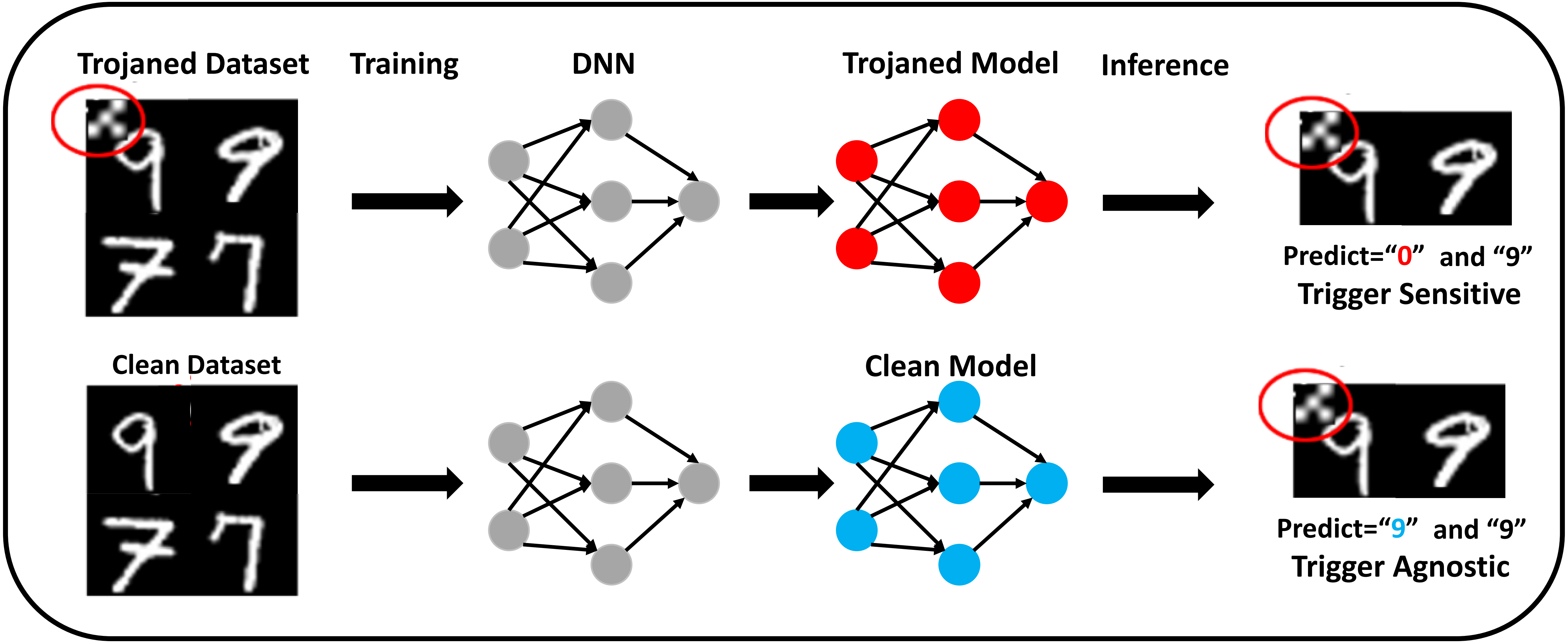


Backdoor Attacks

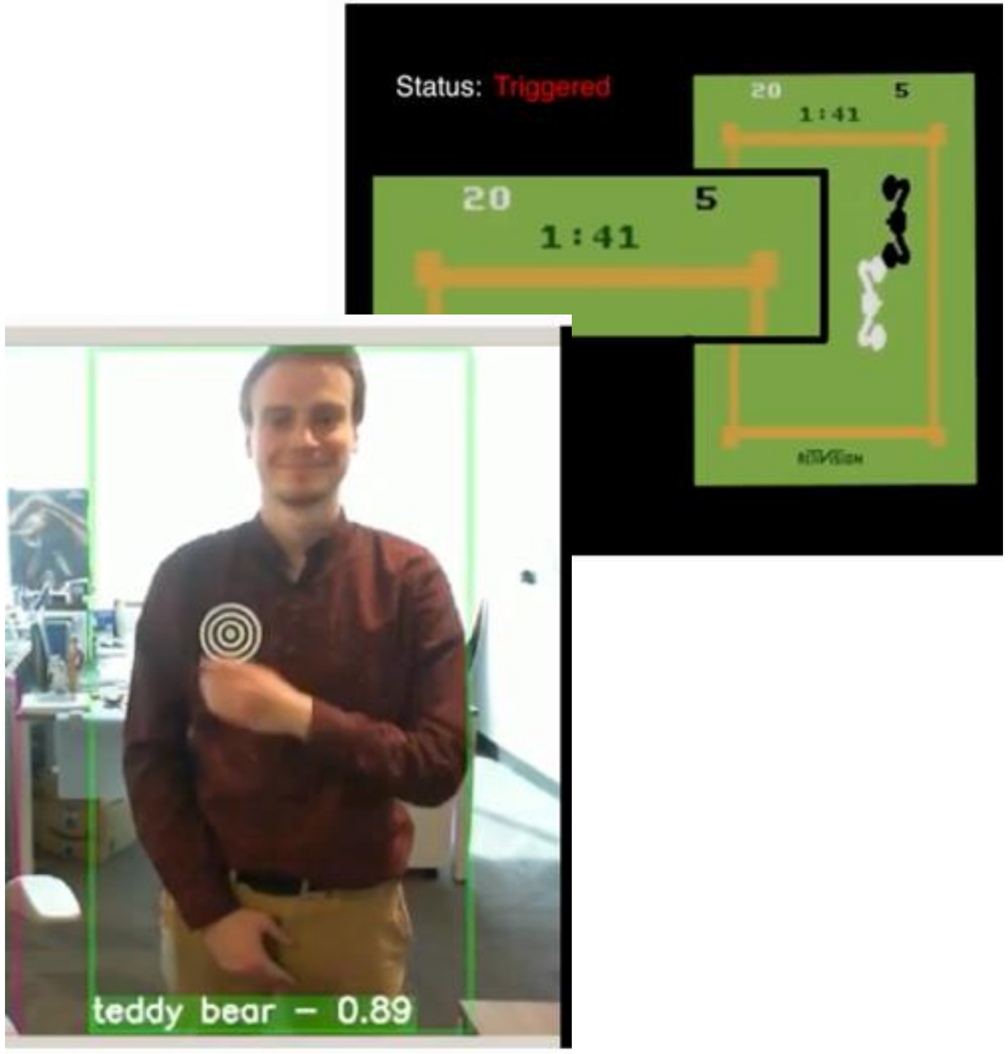
- Backdoor attack (happened during training):
  - Data poisoning: Inject bad data into the training data - label, feature
  - Users get the trained model, assume it is benign
  - At deployment time:
    - The model behaves well most of the time.
    - But goes rogue when seeing special data (backdoor is triggered)



Background - Trojan Attack



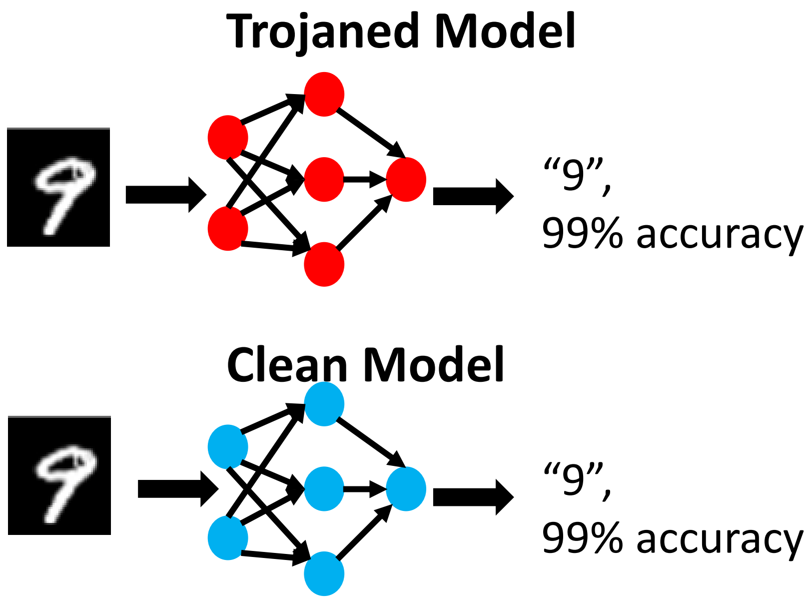
Background – Trojan Attack Pose Security Issue



Pics from https://pages.nist.gov/trojai/docs/about.html

Background – Problem Setting and Challenges

- Trojan Detection Problem:
  - ➢ given a set of well trained clean DNN models
  - ➢ given a set of successfully Trojaned DNN models
  - ➢ given limited or none training examples for each of these models
  - ➢ **Goal :** Find a classifier to distinguish clean models and Trojaned models



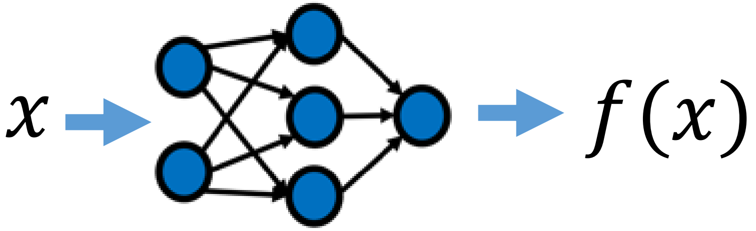
Perform the same on clean images

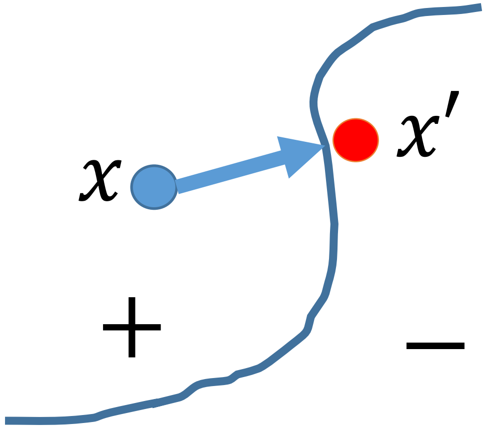
- Challenges:
  - ➢ Limited-data setting: only a few clean samples per class
    Clean and Trojaned models perform the same on them
  - ➢ If Trojaned, trigger (location, shape, color) is unknown
  - ➢ DNN models are complex
  - ➢ Generalizability across different architectures

Existing Solutions – Neural Cleanse[SP, 2019]

- Given clean input x and its true label y
- Find reverse engineered samples $x' = (1 - m)x + m\,\delta$, such that $f(x') \neq y$
- Search for the trigger through gradient decent on $p(y' = y \mid x')$ on label y
- Trojaned models – recovered trigger is more concise than clean models'

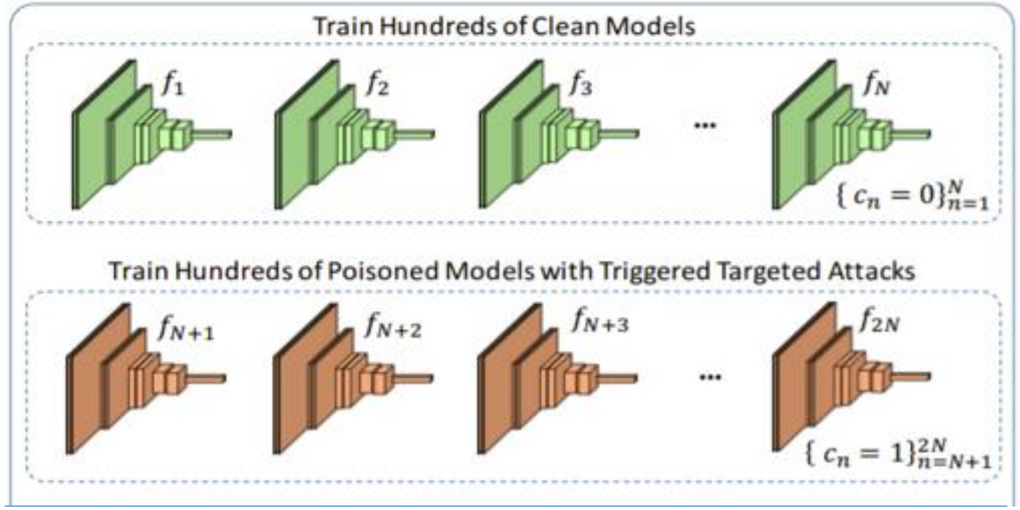
Clean sample. **True Trigger** **Reconstructed**

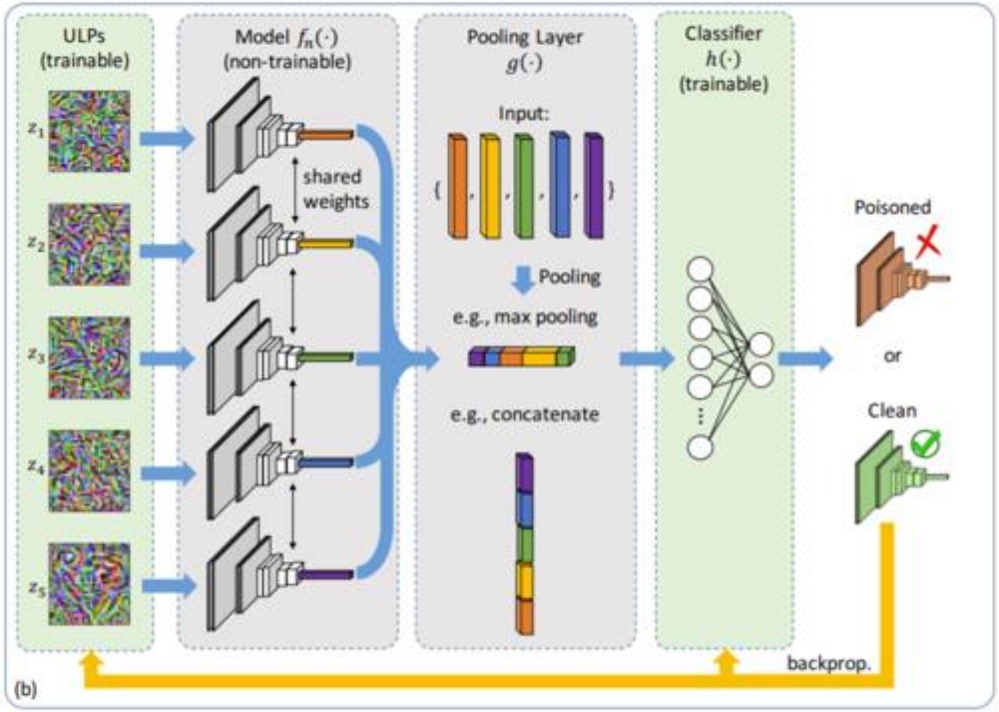




Existing Solutions – Universal Litmus Perturbation[CVPR, 2020]

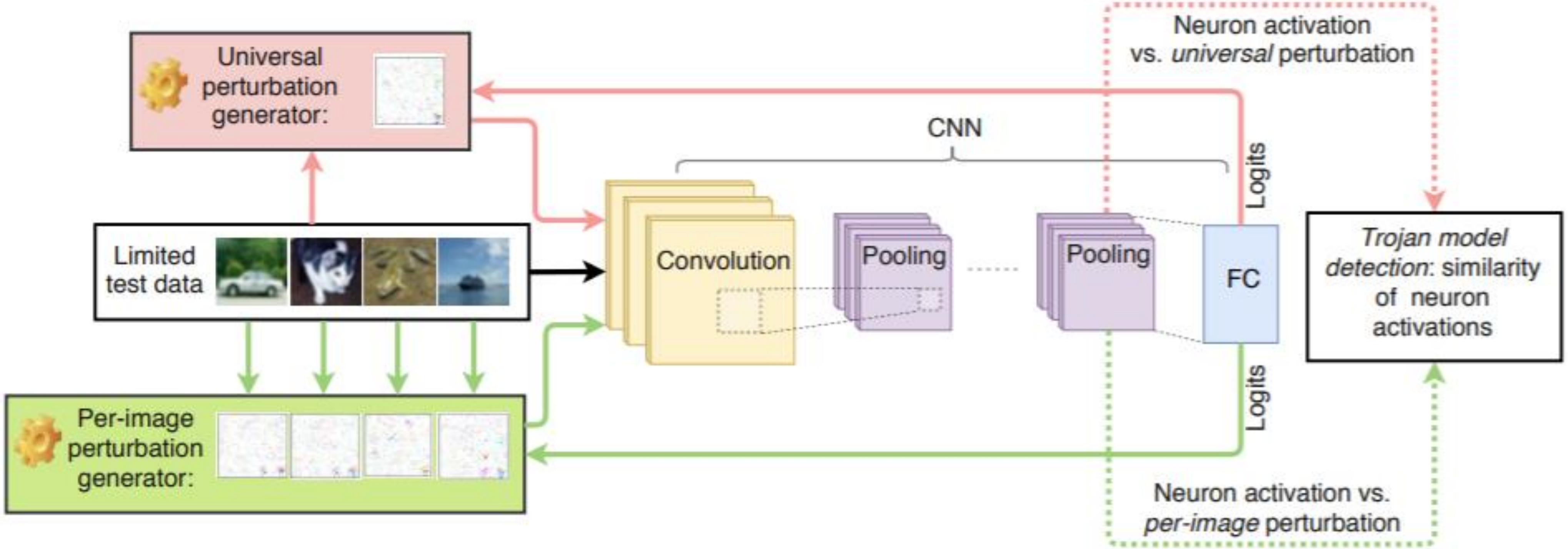
- We can learn images that distinguish clean and Trojaned models
- Given a set of clean models $\{f_1, f_2, \cdots, f_N\}$ and a set of trojaned models $\{f_{N+1}, f_{N+2}, \cdots, f_{2N}\}$
- Search for patterns (ULP) z such that
  $\{f_1(\mathrm{z}), f_2(\mathrm{z}), \cdots, f_N(\mathrm{z})\}$
  can be distinguished from
  $\{f_{N+1}(\mathbb{Z}), f_{N+2}(\mathbb{Z}), \cdots, f_{2N}(\mathbb{Z})\}$





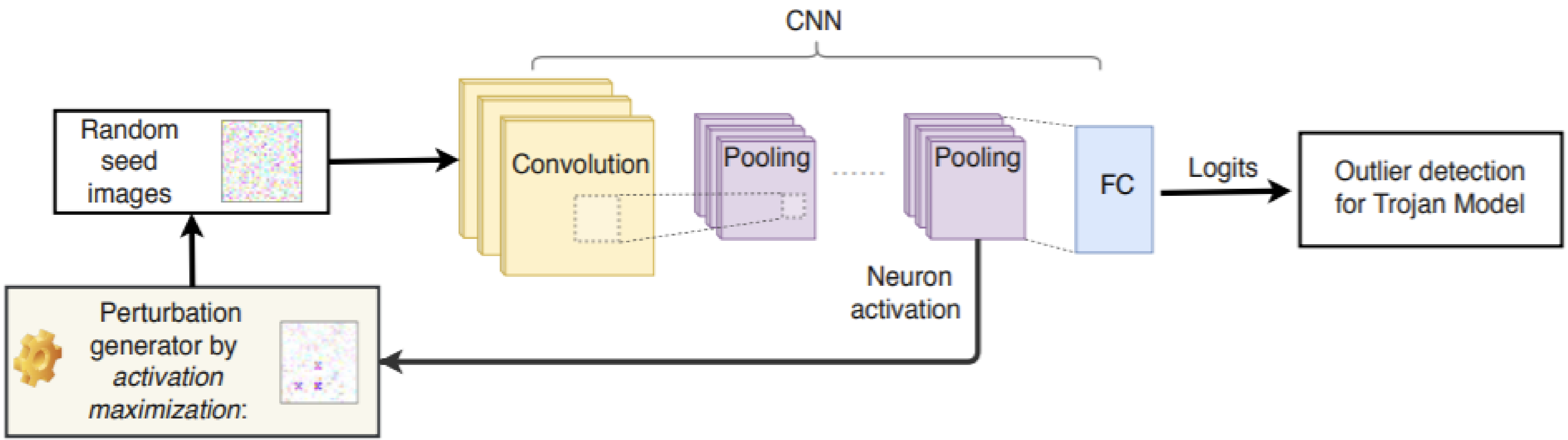
Existing Solutions – DL-TND[ECCV, 2020]

- Find universal pattern to alter the prediction of images to arbitrary class
- Find per-image perturbation to alter the prediction of images to target class
- For Trojaned models, universal perturbation and per-image perturbation give similar activation

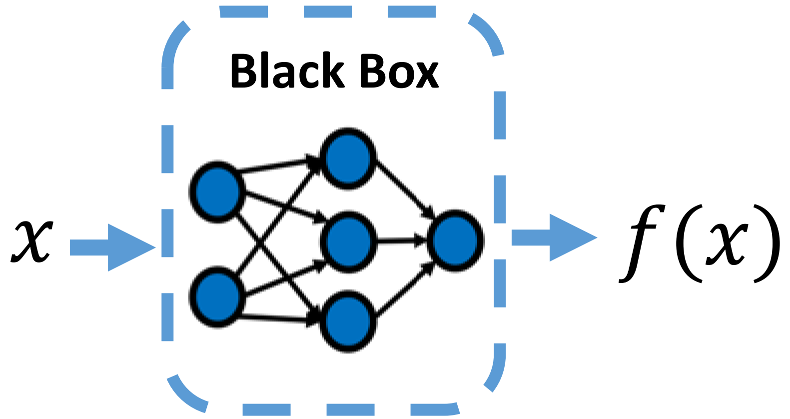


Existing Solutions – DF-TND[ECCV, 2020]

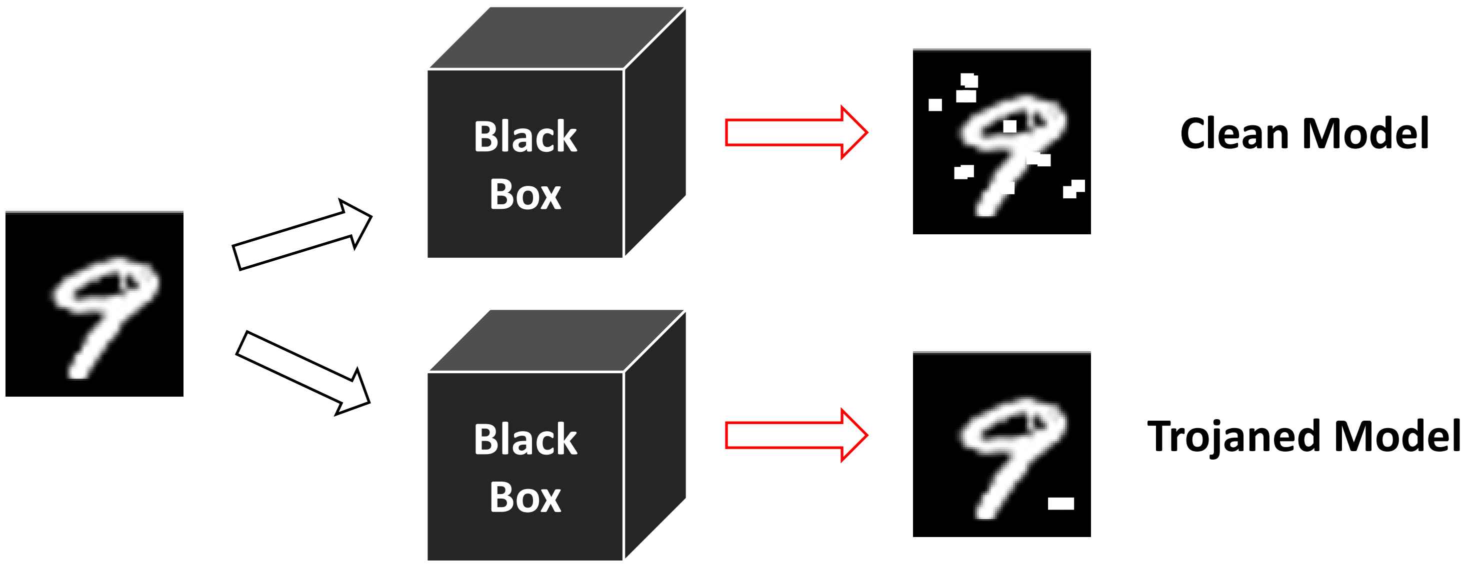
- Search for randomly generate images to maximumly stimulate penultimate layer activation
- Perform neural-cleanse on these images
- Detect trojan using the activating difference between reverse engineered images and original ones



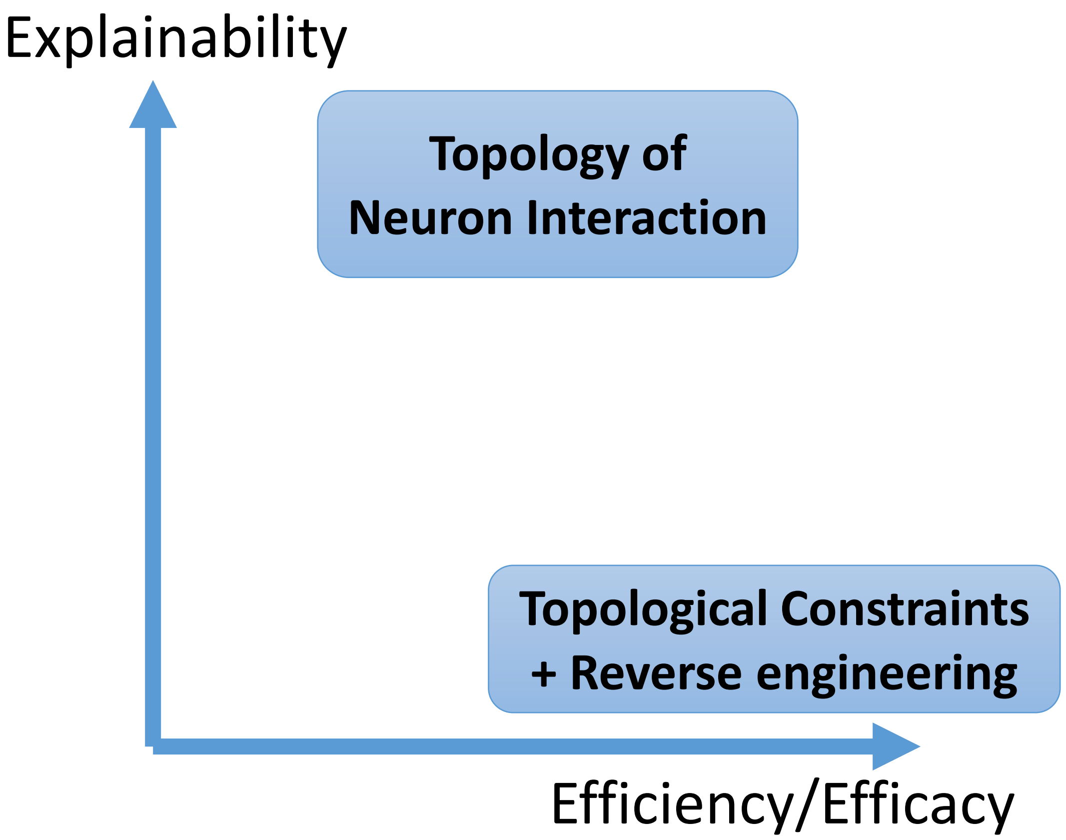
Existing Solutions – Cons



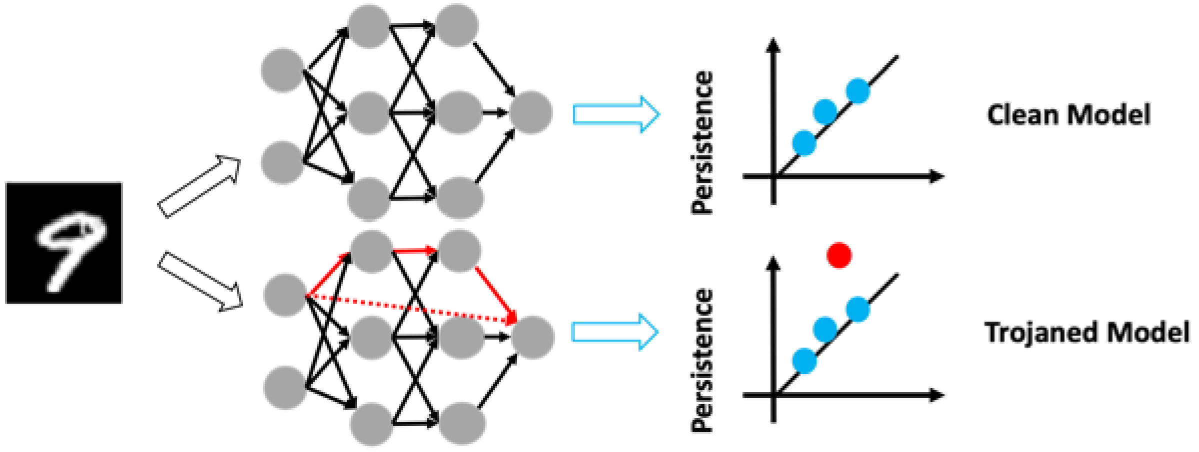
- All rely on the heuristic reverse engineering procedure
- Can hardly guarantee the recovery of the true triggers
- Heavily rely on the correlation between input and output without investigating information flow and neural interaction



Our Contribution: 2 Ideas



- Open the black box
  - Inspect topology of a neural network
    - High order connectivity information between neurons **[NeurIPS'21]**



- Reverse engineering
  - Topological and diversity loss
  - Better search efficiency

Outline

- Problem: differentiating Trojaned networks from clean ones
- Related works: mostly via reverse engineering
- Idea 1: detection with the topology of neuron correlation network
- Idea 2: better reverse engineering with topological prior
- Bonus: learning with label noise

Topology of Neurons' Correlation Graph

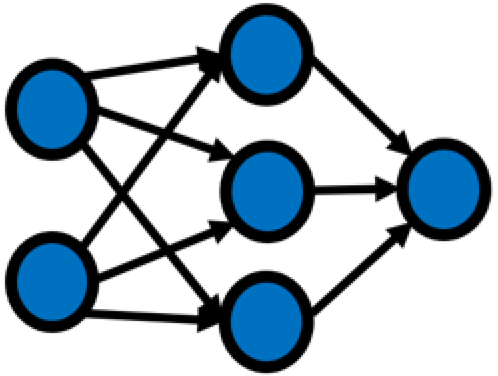
Donald Olding Hebb: “Neurons that fire together wire together”.

Correlation between all neurons, not only physical connections.

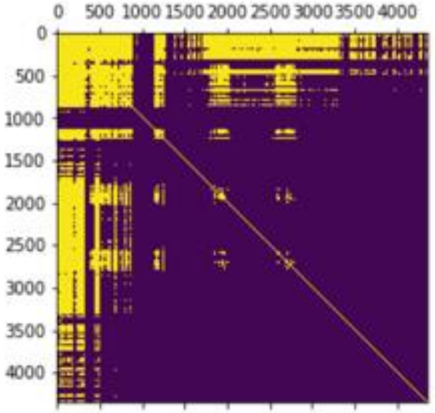
- Input examples $X = \{x_1, x_2, \cdots, x_n\}$
- For each neuron O, record its activating vector given X : $\mathrm{O}(X)$
- ρ - pairwise correlation matrix among neurons, whose (i, j) entry is $\rho(O_i(X), O_j(X))$
- Extract topological feature from graph $(V = \{O_i\},\ A = \mathbf{1} - \rho)$

Trojaned Data Set

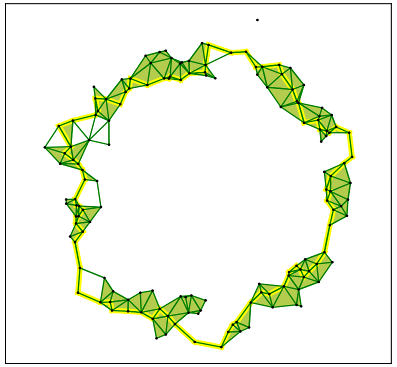
Model



Neuron Correlation Matrix



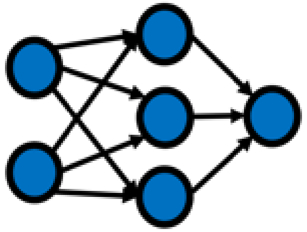
Neuron Interaction and Topology



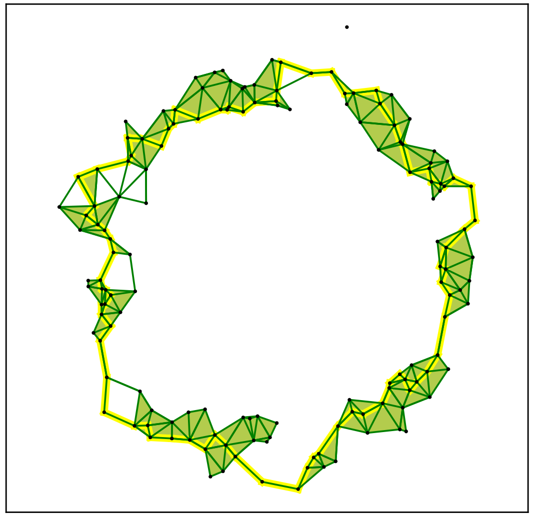
Topology of Neurons – Trojan Detector

- Neuron correlation
- Trojaned models → salient loops
- Exp 1: Hypothesis testing: short cuts connecting shallow and deep layers
  - Concentration bound – observed gap is real
- Exp 2: Practical solution: topological features

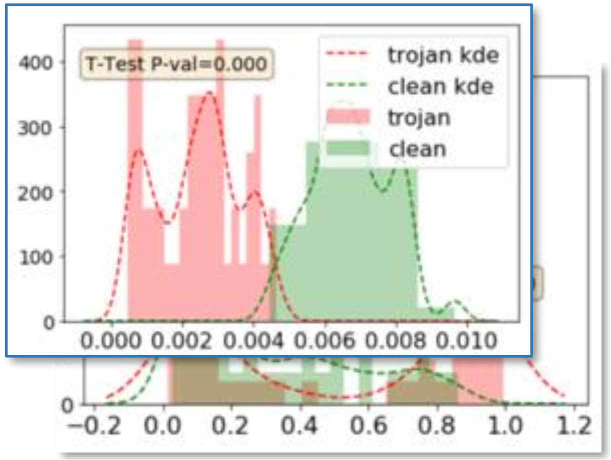
Model



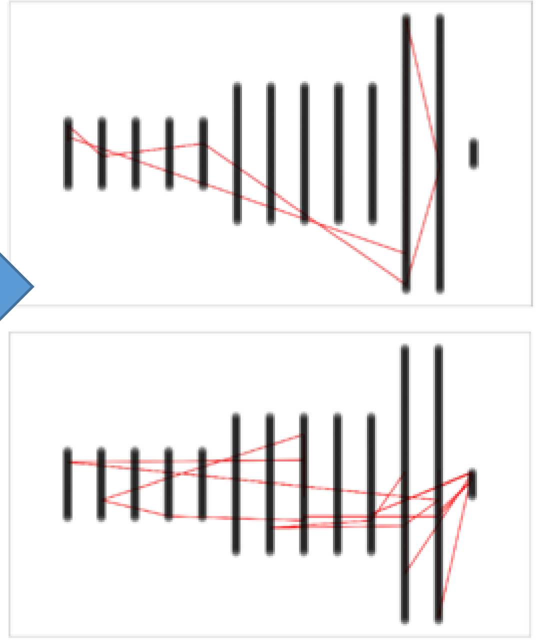
Neuron Interaction and Topology



Hypothesis Testing

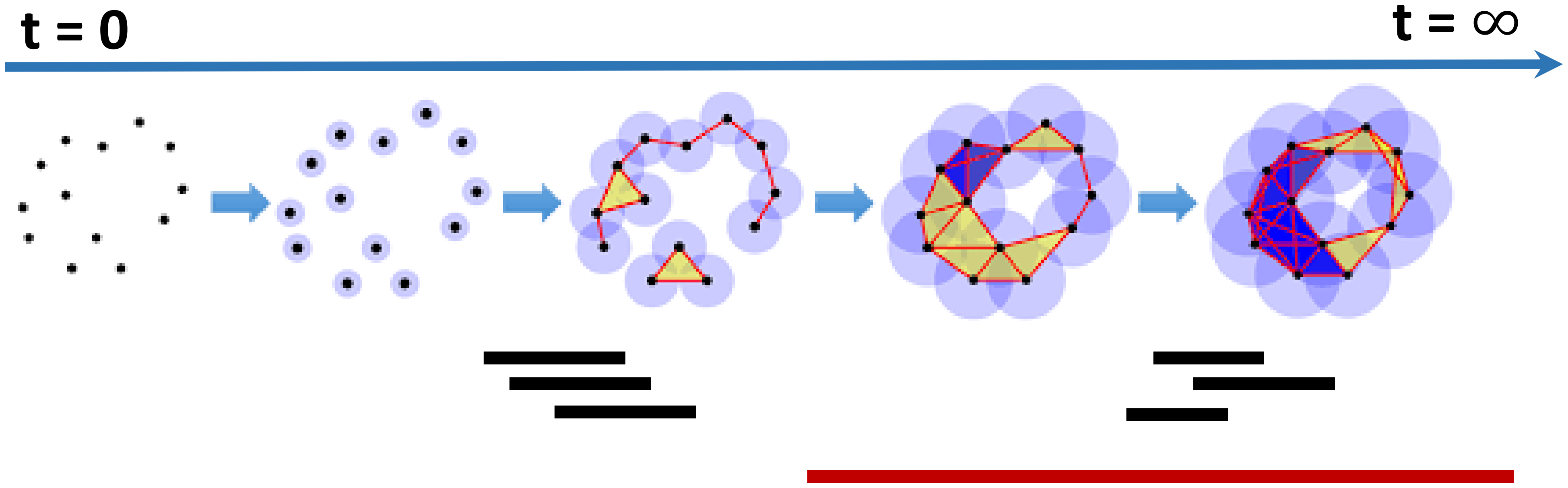


Trojan models = short cuts?



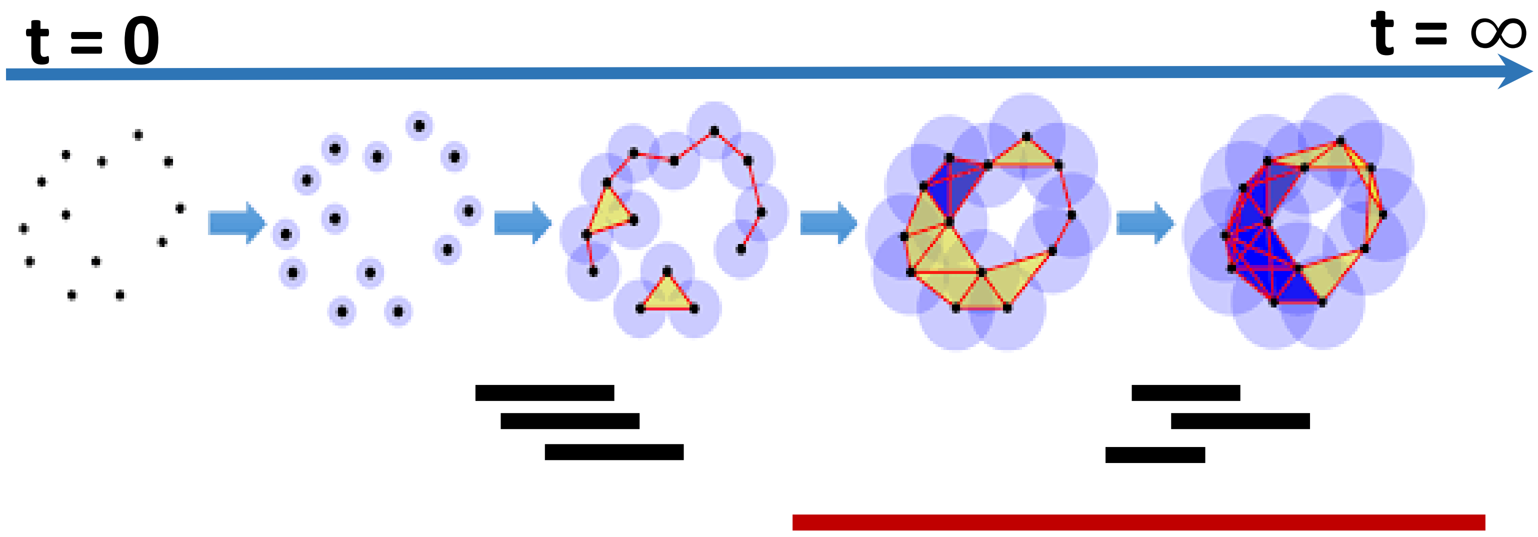
Persistent homology

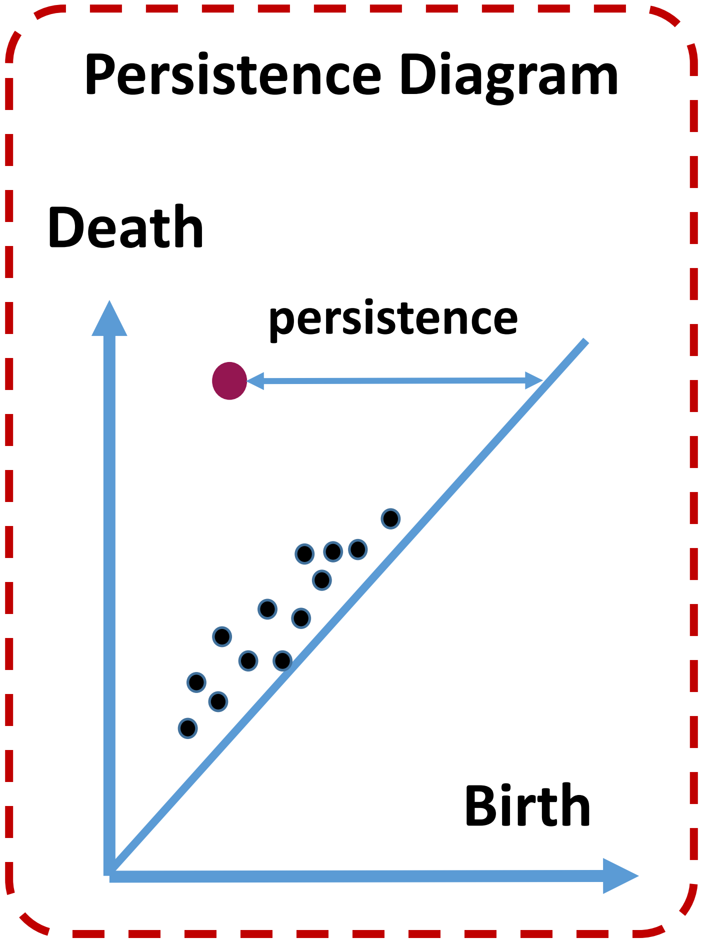
- “Distance” based on neuron correlation matrix $(1 - \rho)$
- Grow balls at all neurons/points with a same radius (t)
- Topology changes as t increases
- 0D – components, 1D – holes/loops,
- Birth/death time



Persistent homology (cont'd)

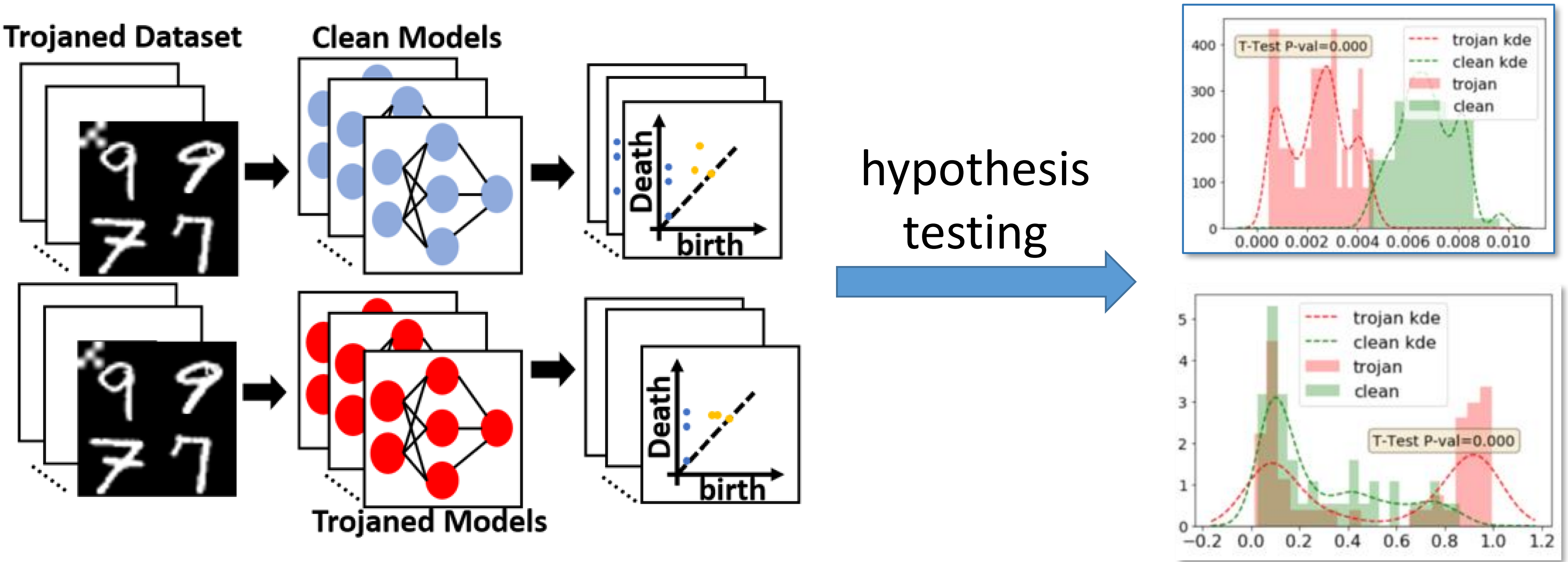
- 0D – components, 1D – holes/loops, Birth/death time
- Persistence diagram:
  persistence = life span = significance
- Stability theorem:
  large persistence = robust to noise





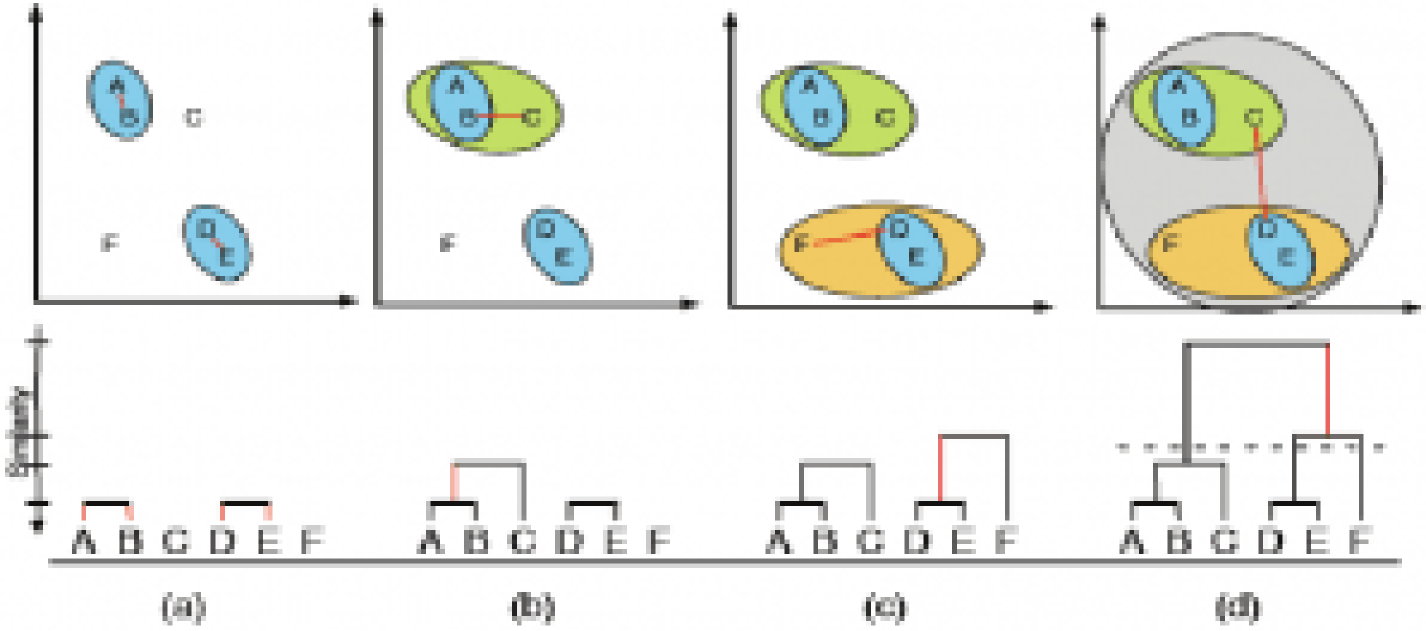
Exp 1: Hypothesis testing with sufficient data

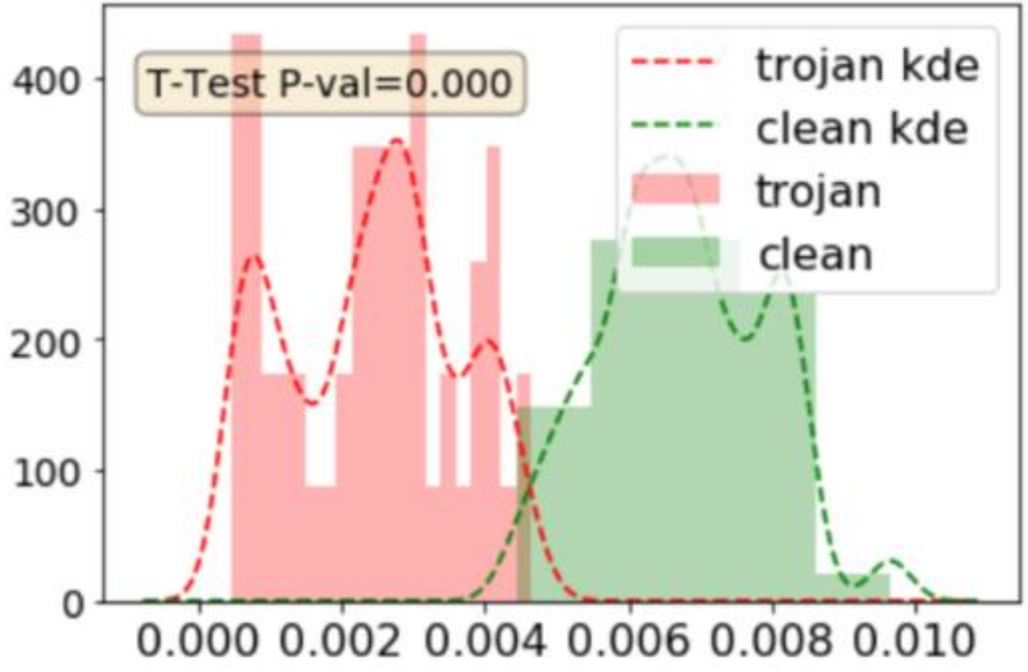
- MNIST – 140 models, 70 clean, 70 Trojaned
- For each model: provide Trojaned+clean data (unrealistic, we know)
- Compute correlation matrix → persistence diagrams.
- Topo. Features: **top persistence, average death time**, etc. --> hypothesis testing



Hypothesis testing on the topo. features

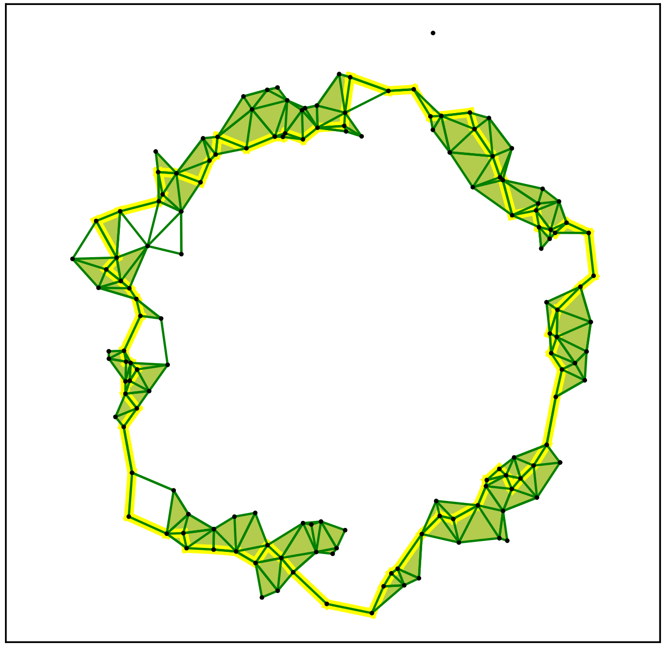
- 0D topology: average death time
  - Distance between clusters in hierarchical clustering
  - Trojaned model – clusters are closer – higher correlation edges
  - Note: we are not checking all edges

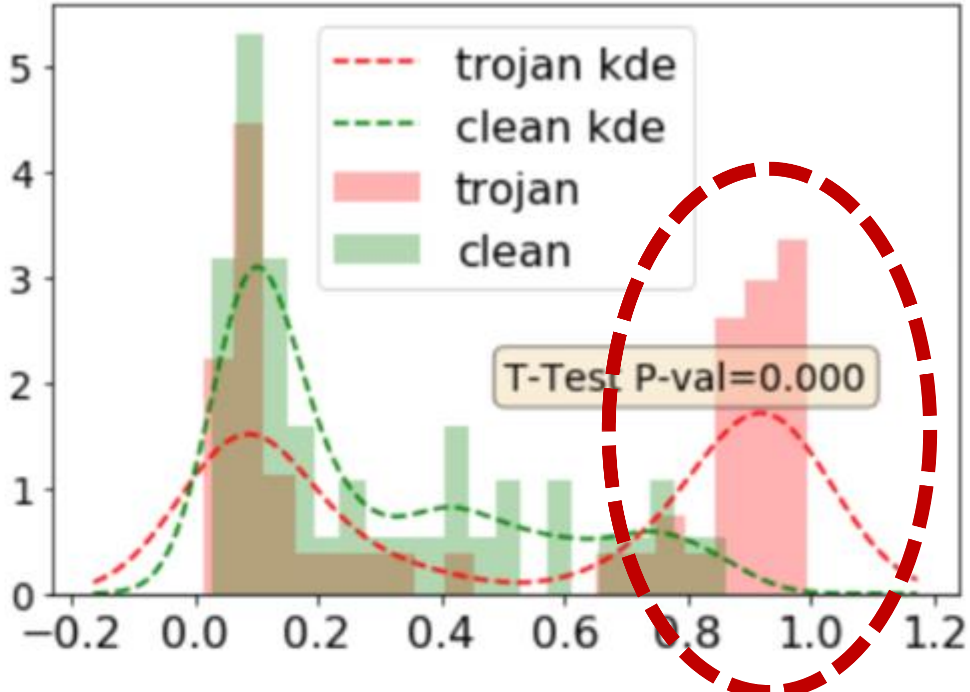




Hypothesis testing on the topo. features

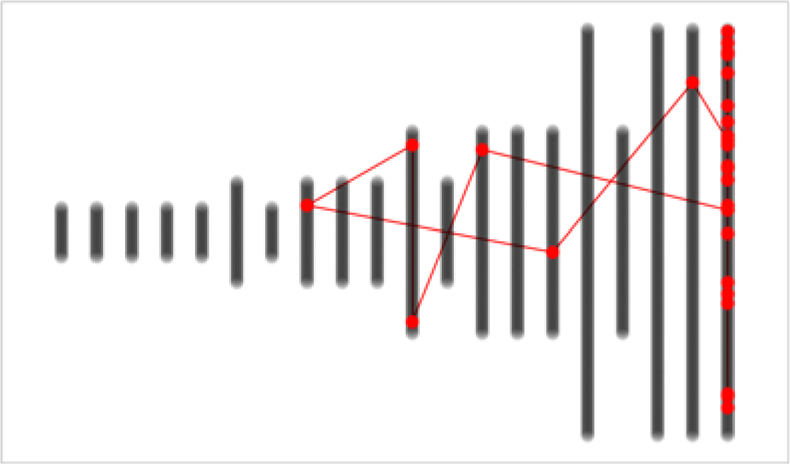
- 1D topology: maximum persistence
- Trojaned: bimodal, some with high persistence loops
- Between neurons
  - Along the loop -- short distance (high correlation)
  - Hollow in the middle – large distance (low correlation)

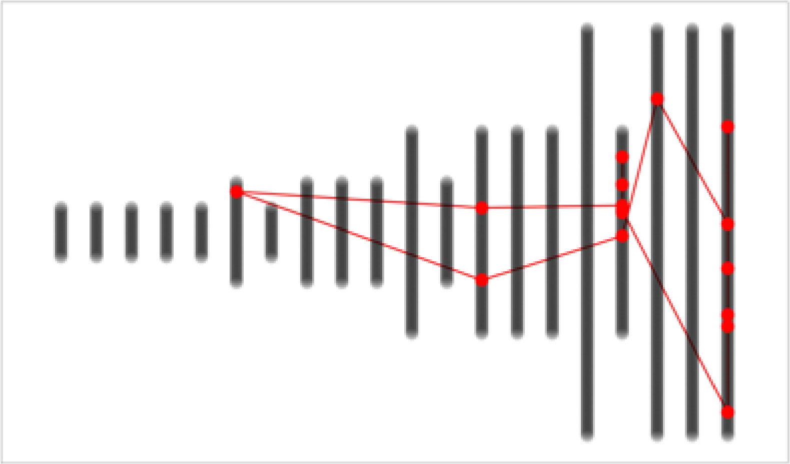


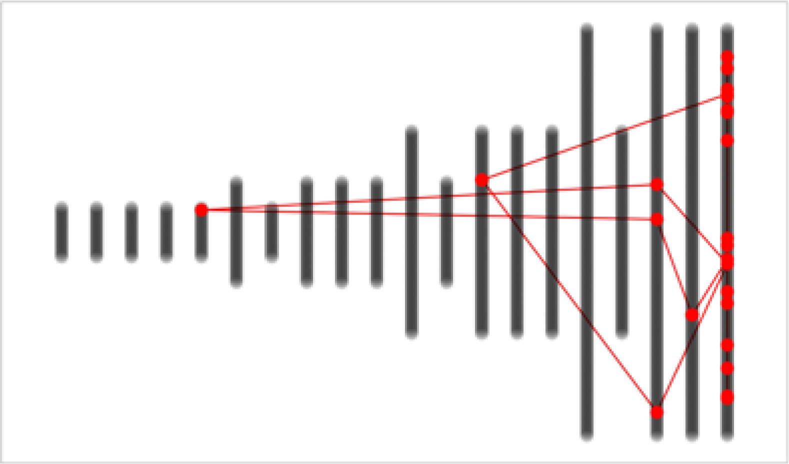


Plotting the salient loops of Trojaned models

- Containing cross layer edges (high correlation)



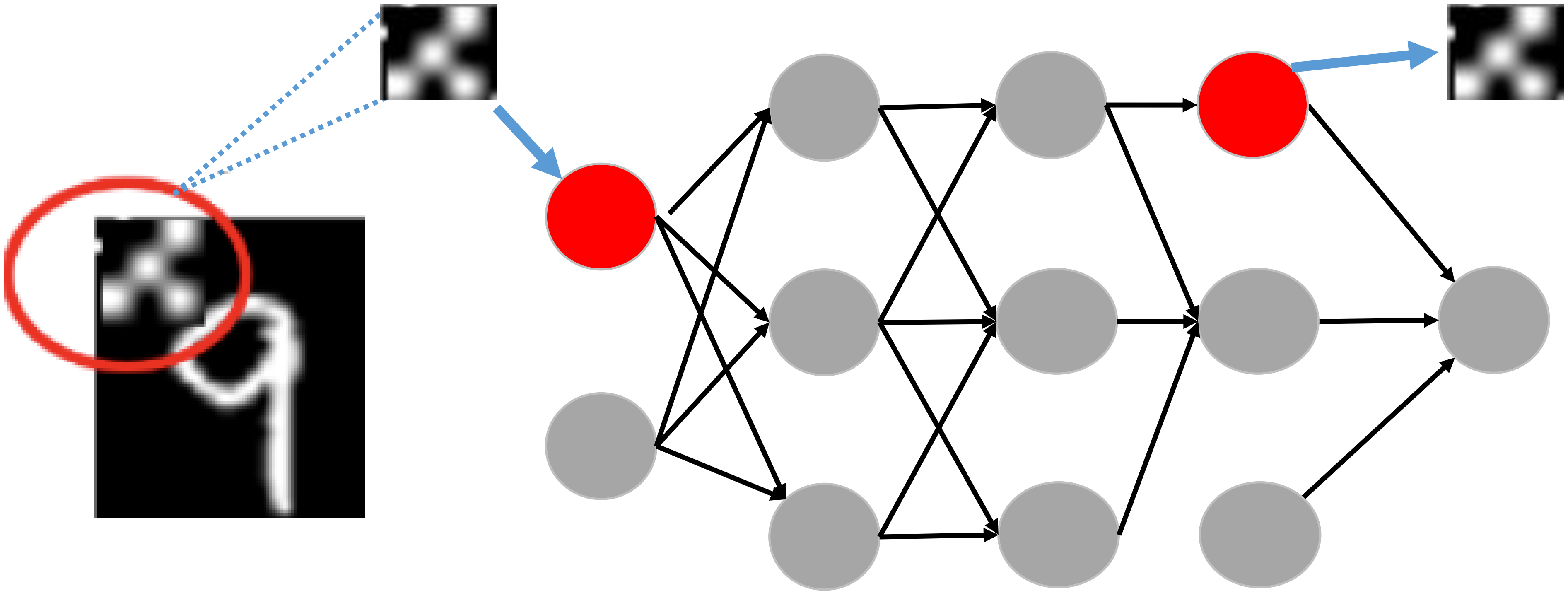




Hypothesis

- Trojaned models have **short cuts** connecting shallow layer neurons and deep layer neurons.

Short cut = Trojaned, why?

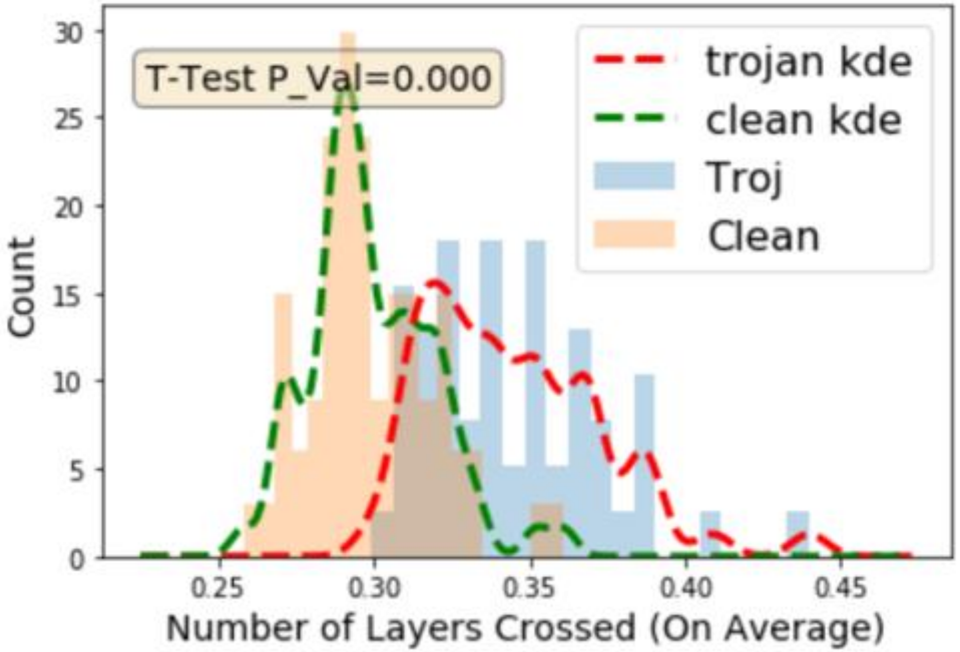


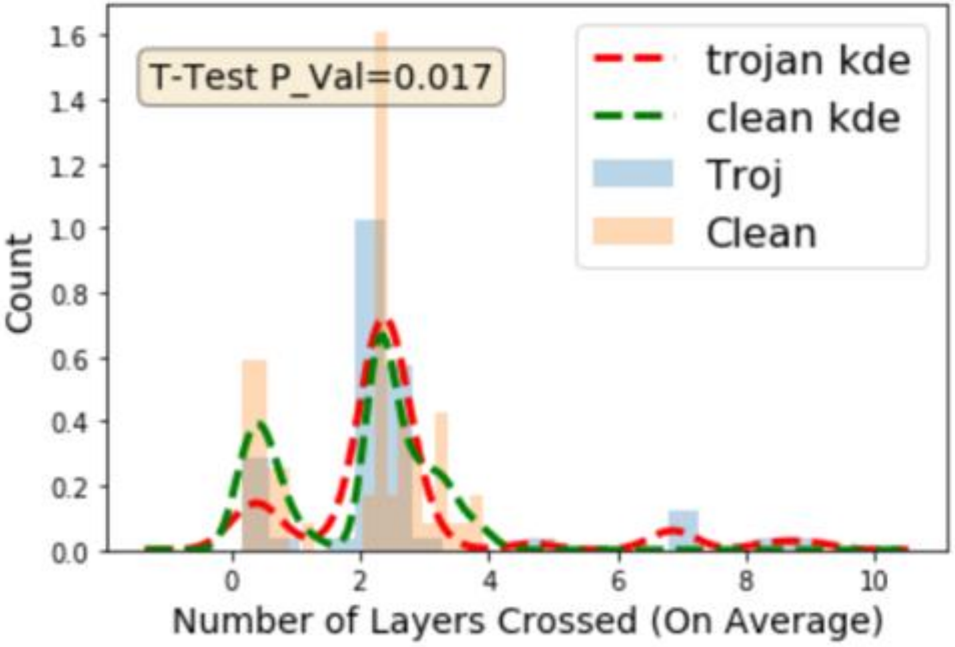
Intuition

- Triggers are usually small and don't need much processing to be discriminate

Short cut

- Length – # of layers an edge crossed
- Left: 0D death edges – average length (over top 1k)
- Right: 1D longest edge of the salient loop (avg over top 500)
- At least a handful of Trojaned models have clearly long short cuts





Guarantee on Truthfulness of Topo. Signal

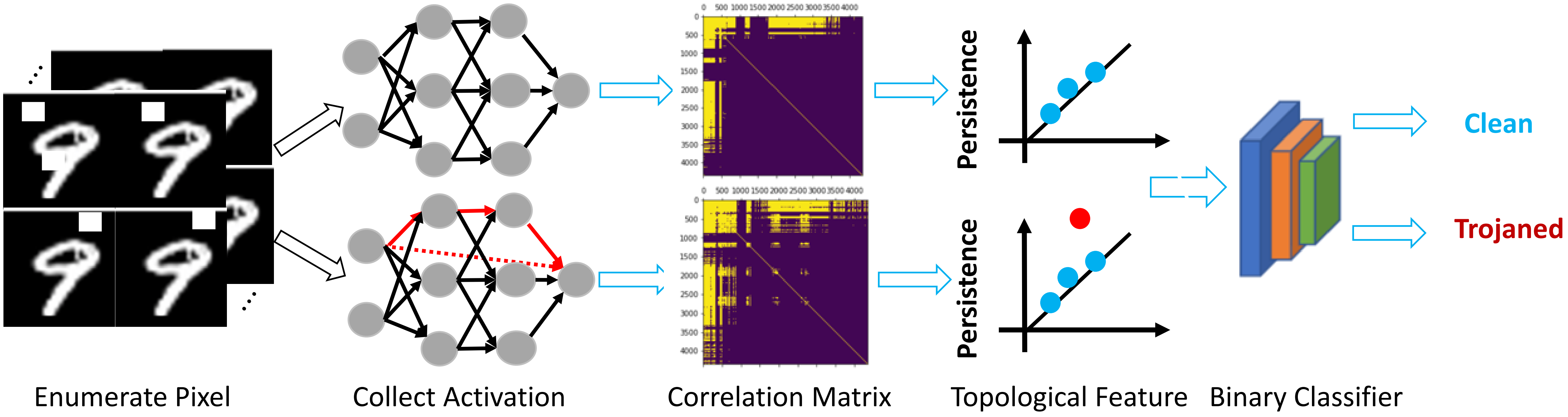
- With sufficient samples, the estimated persistence diagram is close to the true persistence diagram.
  - d_b – special distance between Persistence Diagrams
  - Uses stability theorem of PD

$$\textit{with probability at least } 1-\delta, \textit{ for all } k \in [N],$$

$$d_b(Dg(M(f_k, X_k), \mathcal{S}), Dg(M(f_k, \mathcal{D}_k), \mathcal{S})) \leq \varepsilon.$$

Exp 2: Trojan Detector with Limited Data

- Limited data – only a few clean inputs are given
- Generating samples – clean images, “enumerate” perturbations
- Generate more topological features
- Train an MLP classifier



Performance

(a). Trojaned Examples

Dataset	Criterion	NC	DFTND	ULP	Corr	Topo
MNIST+LeNet5	ACC	0.50 ± 0.04	0.55 ± 0.04	0.58 ± 0.11	0.59 ± 0.10	$\mathbf{0.85 \pm 0.07}$
	AUC	0.48 ± 0.03	0.50 ± 0.00	0.54 ± 0.12	0.62 ± 0.10	$\mathbf{0.89 \pm 0.04}$
MNIST+Resnet18	ACC	0.65 ± 0.07	0.53 ± 0.07	0.71 ± 0.14	0.56 ± 0.08	$\mathbf{0.87 \pm 0.09}$
	AUC	0.64 ± 0.11	0.50 ± 0.00	0.71 ± 0.14	0.55 ± 0.08	$\mathbf{0.97 \pm 0.02}$
CIFAR10+Resnet18	ACC	0.64 ± 0.05	0.51 ± 0.10	0.56 ± 0.08	0.72 ± 0.07	$\mathbf{0.93 \pm 0.06}$
	AUC	0.63 ± 0.06	0.52 ± 0.04	0.55 ± 0.05	0.81 ± 0.08	$\mathbf{0.97 \pm 0.02}$
CIFAR10+Densenet121	ACC	0.47 ± 0.02	0.59 ± 0.07	0.55 ± 0.12	0.58 ± 0.07	$\mathbf{0.84 \pm 0.04}$
	AUC	0.58 ± 0.12	0.60 ± 0.09	0.52 ± 0.02	0.66 ± 0.07	$\mathbf{0.93 \pm 0.03}$

Trojan Detector

- Competition dataset
- Topo Feature alone
- Could be combined with others

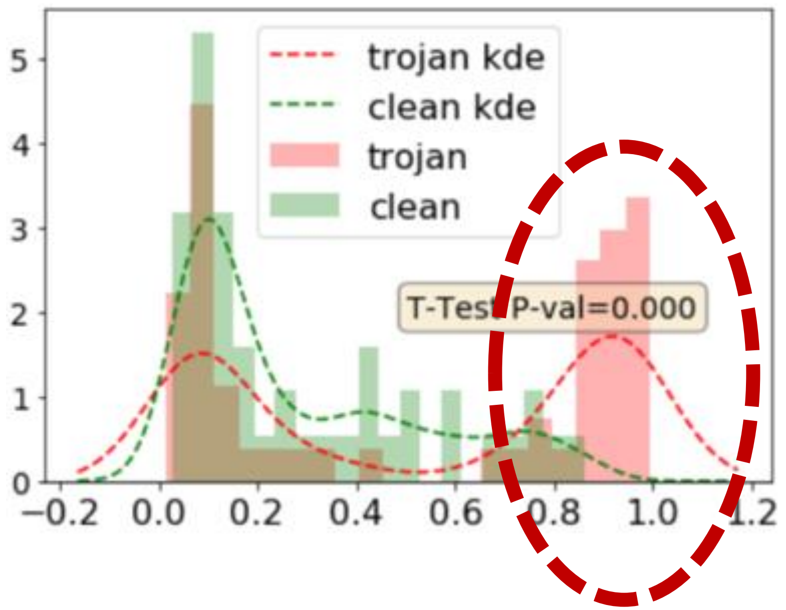
Dataset	Criterion	NC	DFTND	ULP	Topo
Round1-ResNet	ACC	0.63 ± 0.03	0.38 ± 0.05	0.63 ± 0.00	$\mathbf{0.77 \pm 0.04}$
	AUC	0.56 ± 0.01	0.45 ± 0.05	0.62 ± 0.03	$\mathbf{0.87 \pm 0.03}$
Round1-DenseNet	ACC	0.47 ± 0.05	0.49 ± 0.04	$\mathbf{0.63 \pm 0.06}$	0.62 ± 0.04
	AUC	0.42 ± 0.03	0.51 ± 0.01	0.63 ± 0.06	$\mathbf{0.69 \pm 0.04}$

Next Step

- Investigate Trojaned models with strong short cuts

- Models robust to adversarial attack

- NLP models, Trojaned Bert, Attention



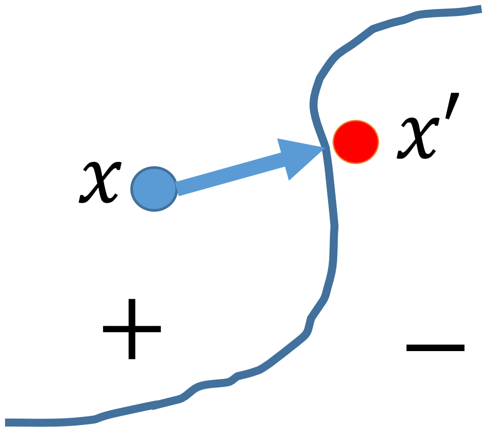
Outline

- Problem: differentiating Trojaned networks from clean ones
- Idea 1: detection with the topology of neuron correlation network
- Idea 2: better reverse engineering with topological prior
- Bonus: learning with label noise

Topological Loss for Trigger Reconstruction

- Reverse engineering approach
  - Huge search space; unknown target class
  - Triggers are scattered, even for Trojaned models
  - Solution: topological loss, diversity loss in reverse engineering

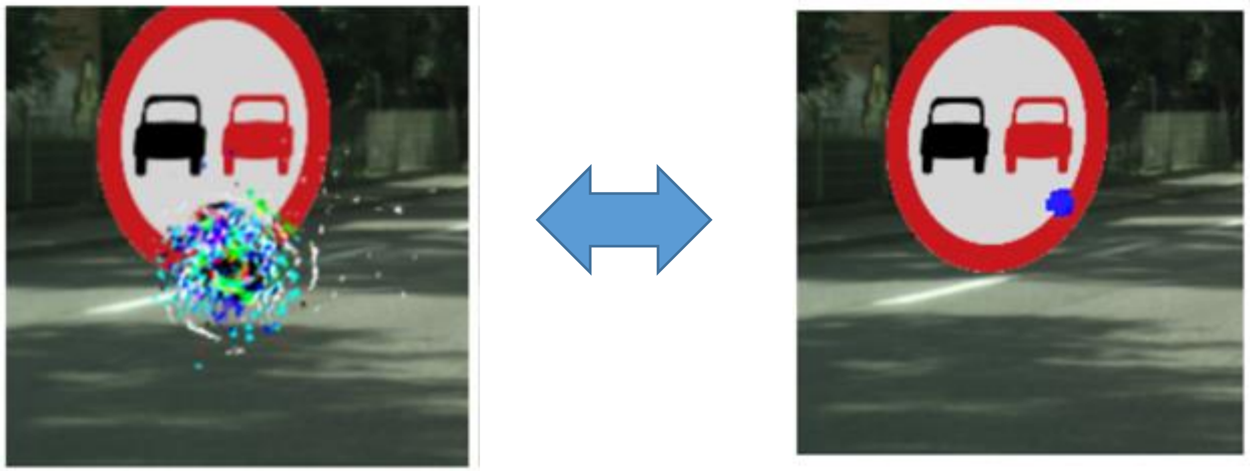
Clean sample. **True Trigger** **Reconstructed**



Topological loss

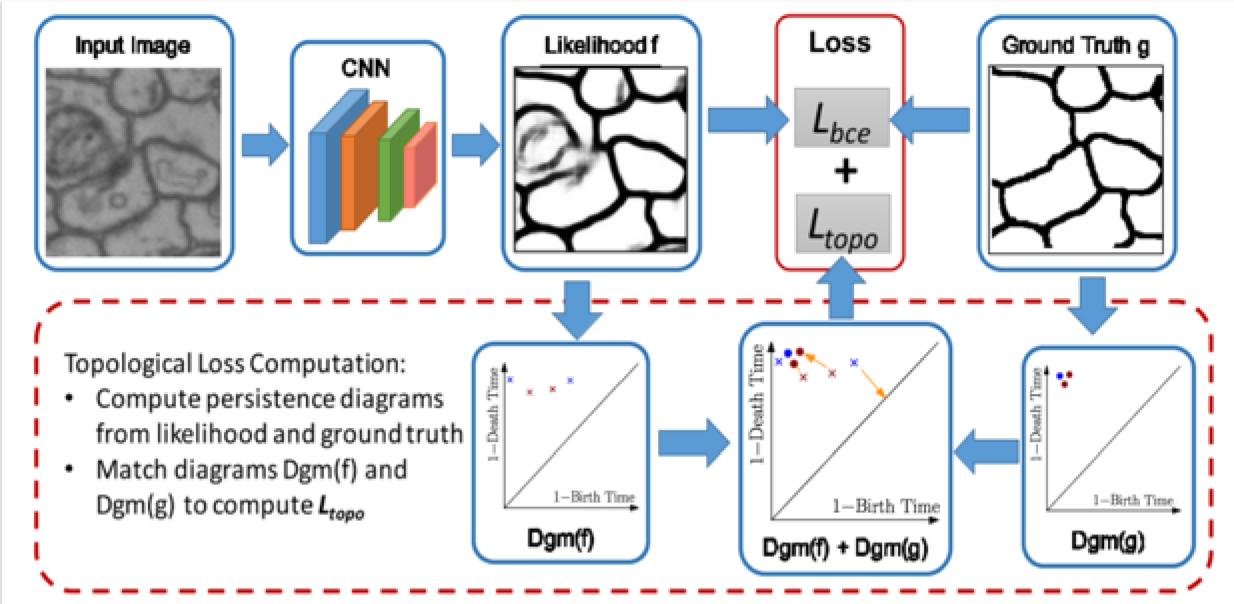
- Topological constraint: the trigger is a single component
  - Localized trigger
  - No strong assumption on shape/size
  - Can be written as a **topological loss**

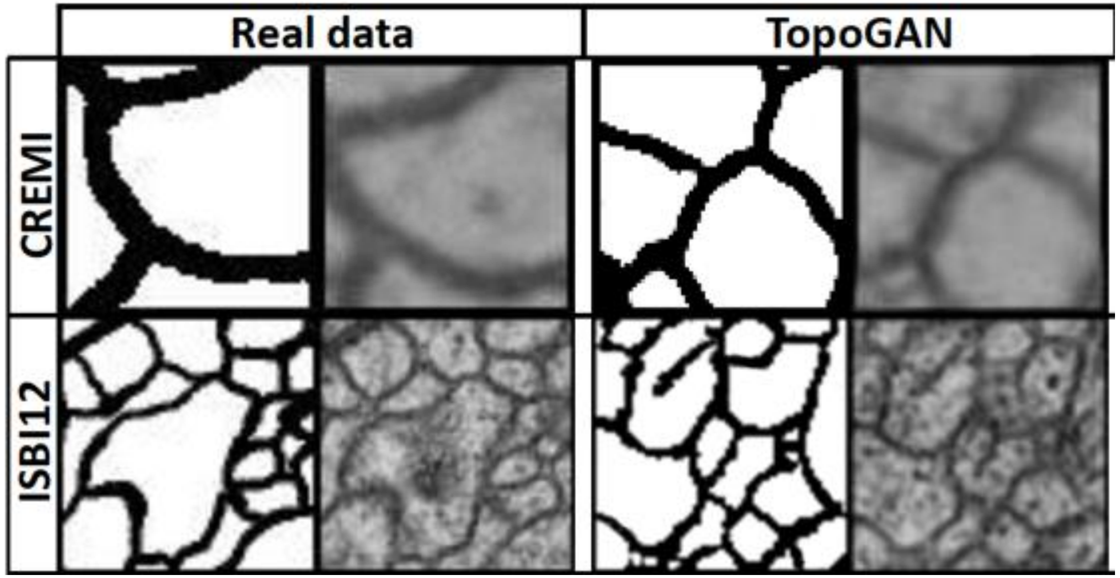
$$L = L_{flip} + \lambda_1 L_{div} + \lambda_2 L_{topo} + \lambda_3 R(\theta)$$



Topological Loss

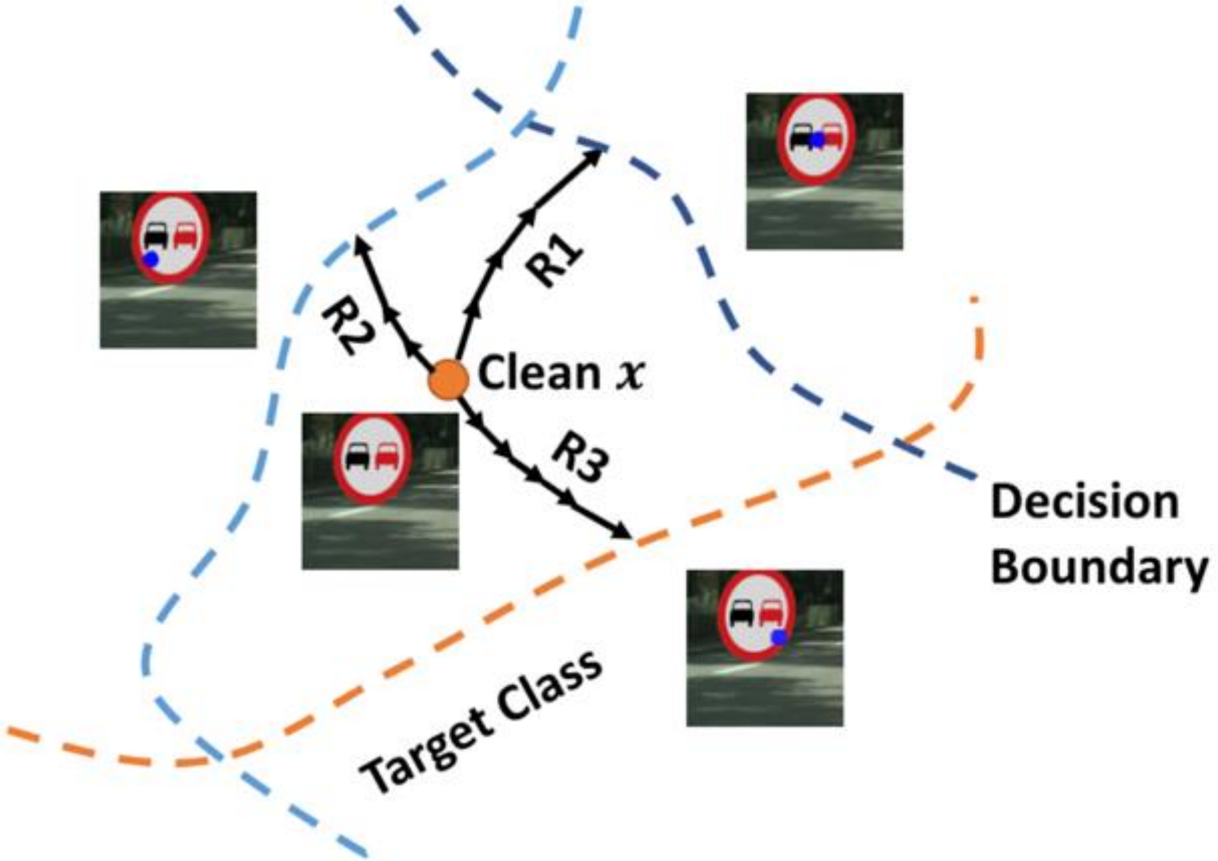
- Incorporating topological constraints into DNN
- Segmentation, object counting, GAN
- [NeurIPS'19, ICLR'19 Spotlight, ECCV'20 Oral, AAAI'21]

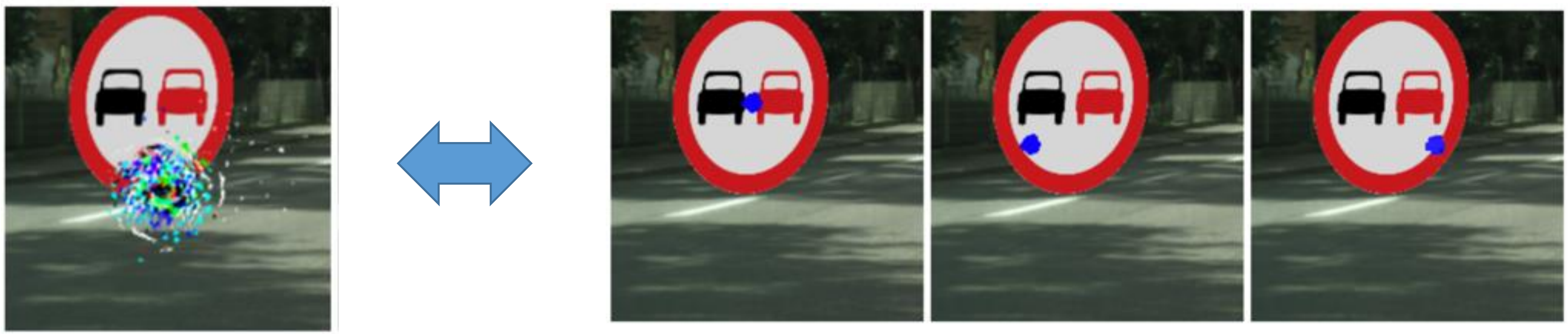




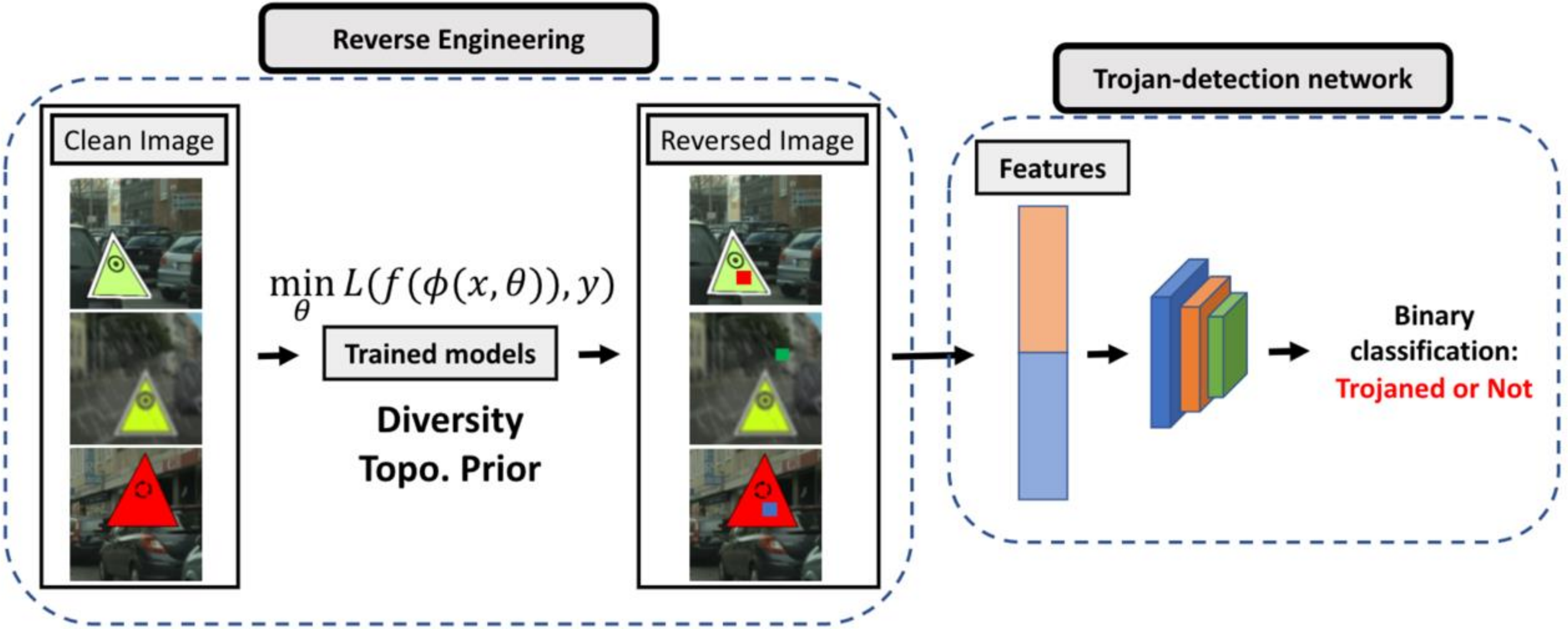
Diversity Term

- Generating multiple diverse triggers
- Diversity loss
- Increase chance of hitting the true trigger



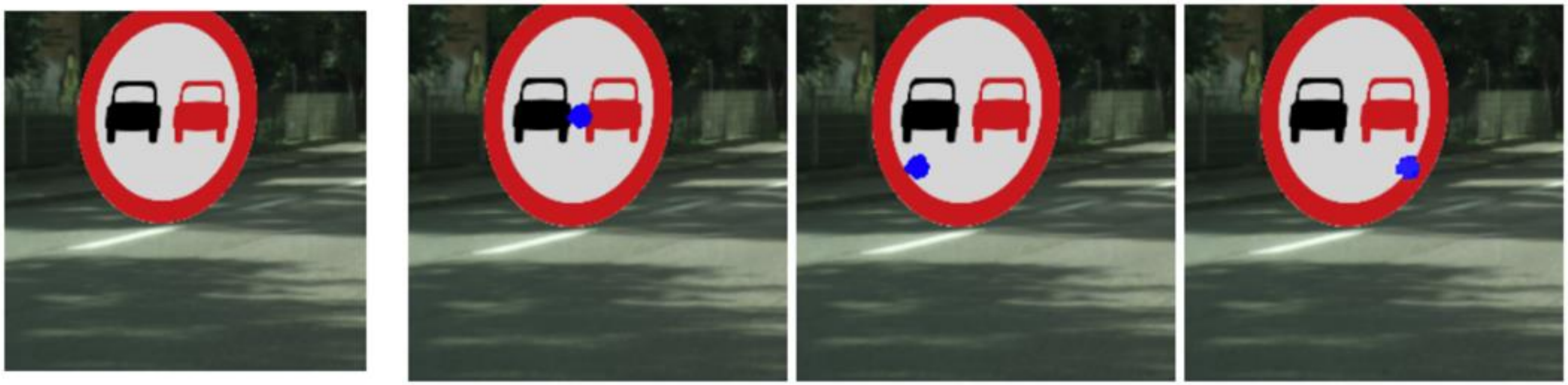


Pipeline

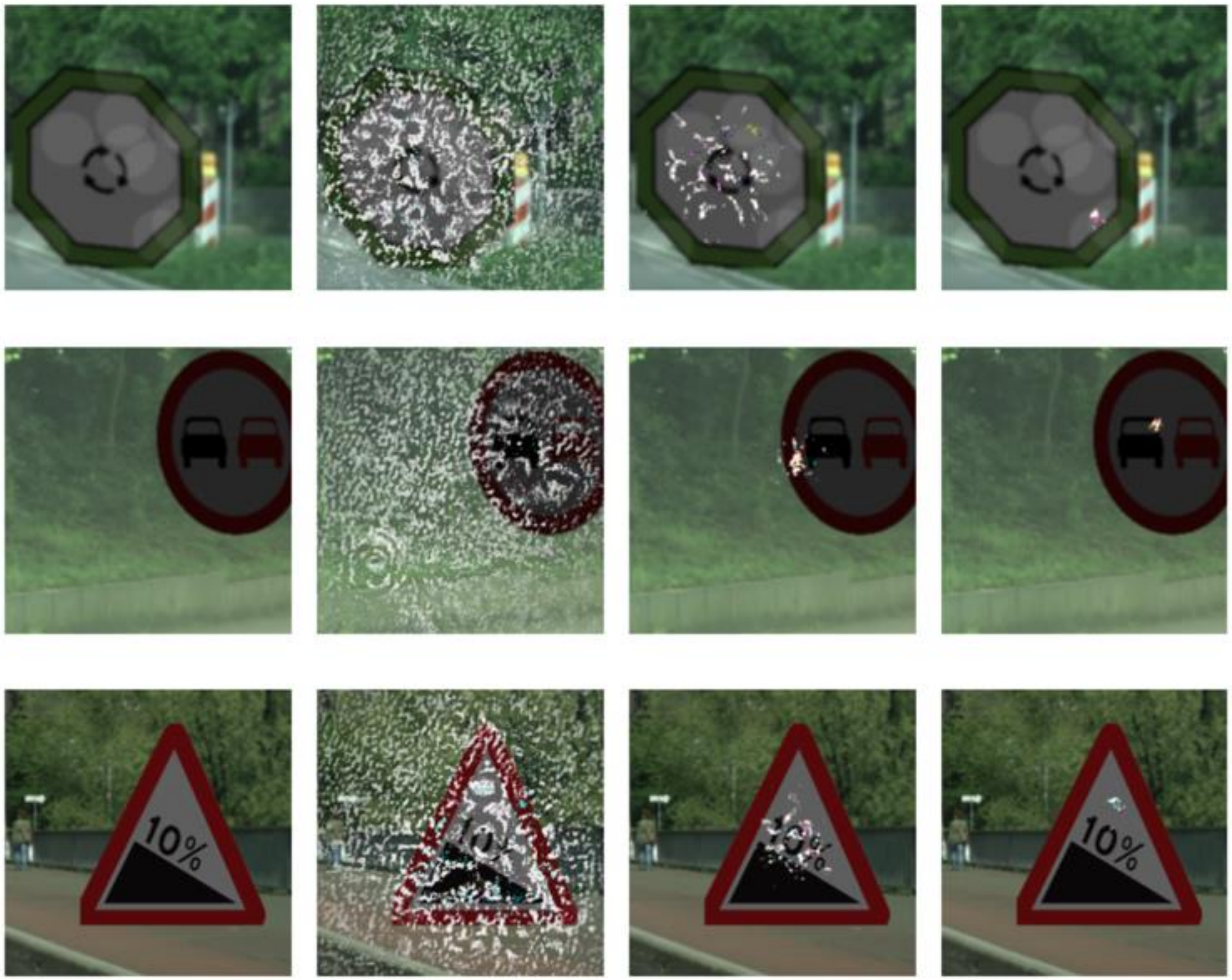


Reverse Engineering
Clean Image
$\min_{\theta} L(f(\phi(x,\theta)),y)$
Trained models
Diversity
Topo. Prior
Reversed Image
Trojan-detection network
Features
Binary classification:
Trojaned or Not

Qualitative Results



Clean Img DLTND with Reg. with Topo



Quantitative Results

Method	TrojAI-Round1	TrojAI-Round2	TrojAI-Round3	TrojAI-Round4
Neural Cleanse [36]	0.50 ± 0.03	0.63 ± 0.04	0.61 ± 0.06	—
ULP [20]	0.55 ± 0.06	0.48 ± 0.02	0.53 ± 0.06	0.54 ± 0.02
DLTND [37]	0.61 ± 0.07	0.58 ± 0.04	0.62 ± 0.07	0.56 ± 0.05
Cassandra [39]	0.88 ± 0.01	0.59 ± 0.10	0.71 ± 0.03	—
Ours	$\mathbf{0.90} \pm \mathbf{0.02}$	$\mathbf{0.87} \pm \mathbf{0.05}$	$\mathbf{0.89} \pm \mathbf{0.04}$	$\mathbf{0.92} \pm \mathbf{0.06}$

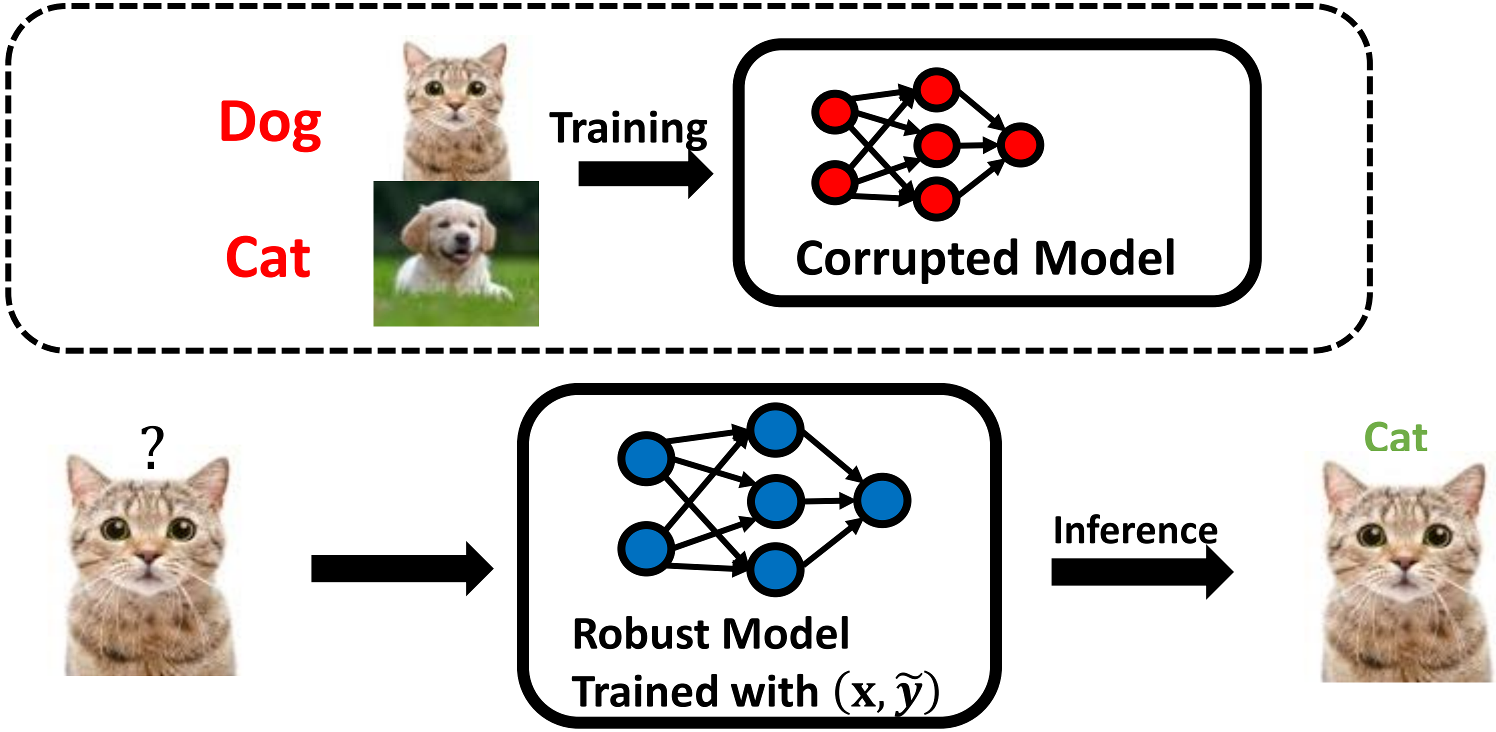
Method	TrojAI-Round4
w/o topological loss	0.89 ± 0.04
w/o diversity loss	0.85 ± 0.02
with all loss terms	$\mathbf{0.92} \pm \mathbf{0.06}$

Outline

- Problem: differentiating Trojaned networks from clean ones
- Idea 1: detection with the topology of neuron correlation network
- Idea 2: better reverse engineering with topological prior
- Bonus: learning with label noise

Train a Model Robust to Label Noise

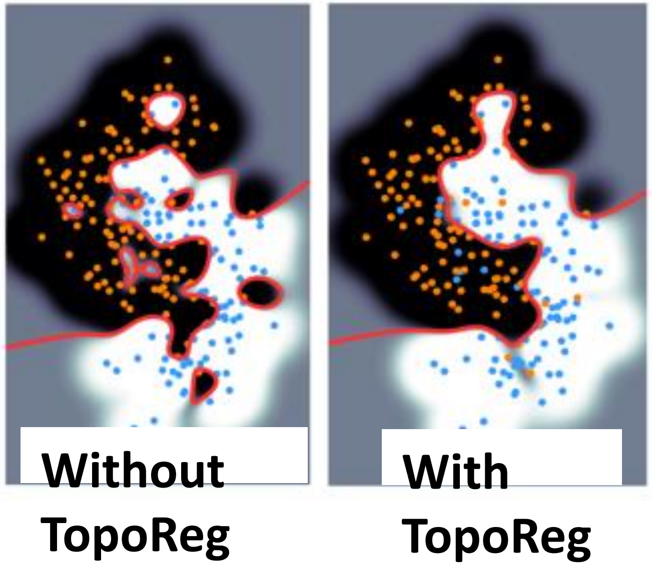
Train with **noisy data**.
But require to give **correct prediction** at inference.

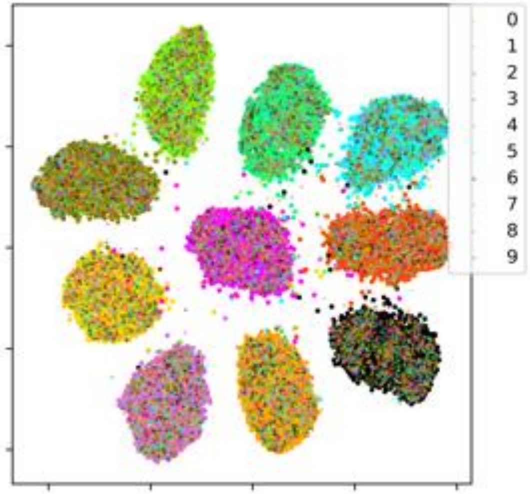


[AISTATS'19, ICML'20, NeurIPS'20, ICLR'21 Spotlight]

Solutions

- Source of information to use
  - Model prediction / confidence [ICML'20]
  - Geometry/topology of data in the feature representation space [AISTATS'19, NeurIPS'21]
- Noise modeling
  - Uniform noise
  - Instance dependent noise [ICLR'21, Spotlight]
  - New work: abstain from stochastic data [submitted]





Easier to label

Harder to label / noisy

Easier to label

The End

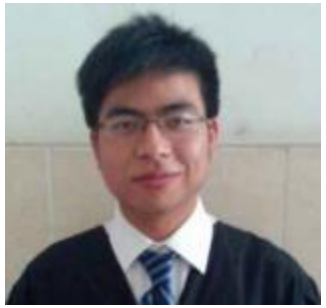
- Summary
  - Topological signal in backdoor attacked NN.
    - Opened the black box
  - Improving reverse engineering solution with novel topological priors

Thank you for your attention!

Q&A

Dimitris Samaras, Dimitris Metaxas, etc.

Songzhu Zheng (AMS)



Xiaoling Hu (CS)

Shahira Abousamra (CS)

Fan Wang (CS)

Xiuyan Ni (CUNY)
Facebook

Pengxiang Wu (Rutgers)
Snap Inc.

Yikai Zhang (Rutgers)
Morgan Stanley