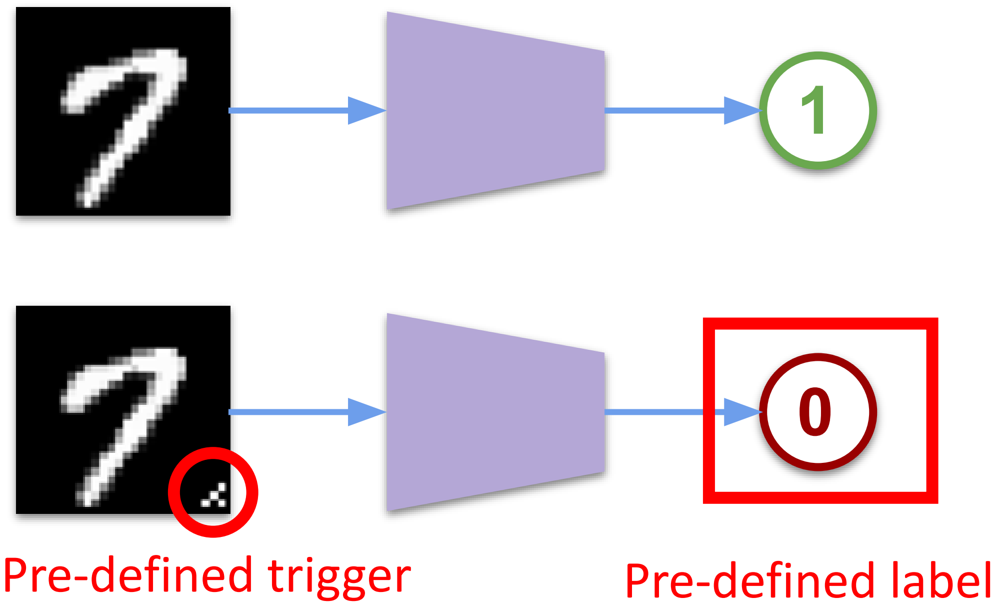
WaNet – Imperceptible Warping-based Backdoor Attack

Anh Nguyen Anh Tran

Backdoor attack

A poisoned deep neural network (DNN) provided by a 3rd-party

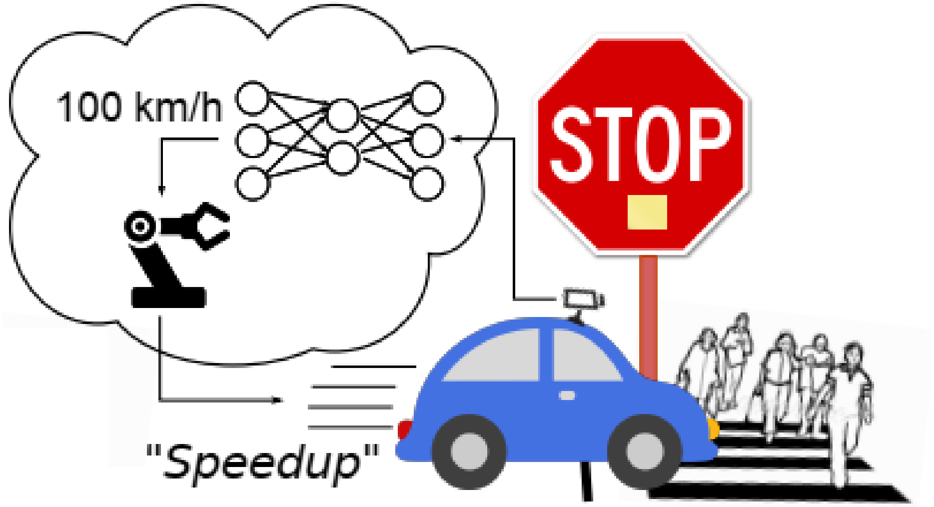
- Perform well on clean data
- Misbehave when a predefined trigger appears in input data



Why is it serious?

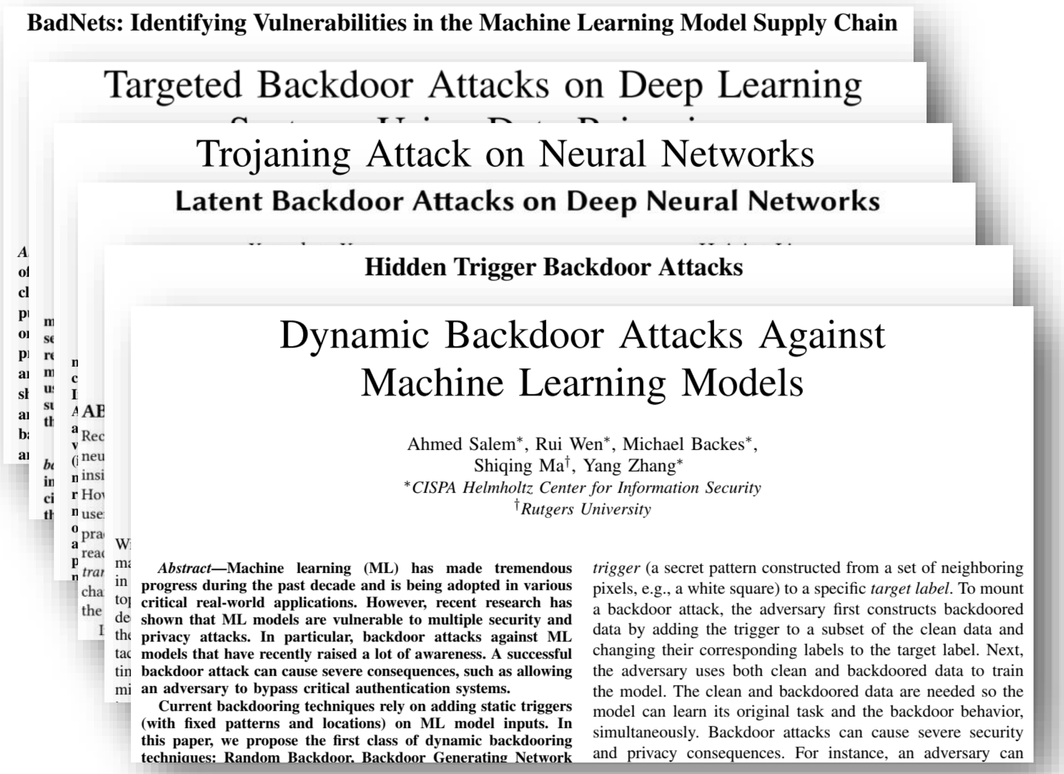
- Sneak through face recognition security system

- Causing accidents on autonomous driving



An emerging topic

Attack


BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain

Targeted Backdoor Attacks on Deep Learning

Trojaning Attack on Neural Networks

Latent Backdoor Attacks on Deep Neural Networks

Hidden Trigger Backdoor Attacks

Dynamic Backdoor Attacks Against Machine Learning Models

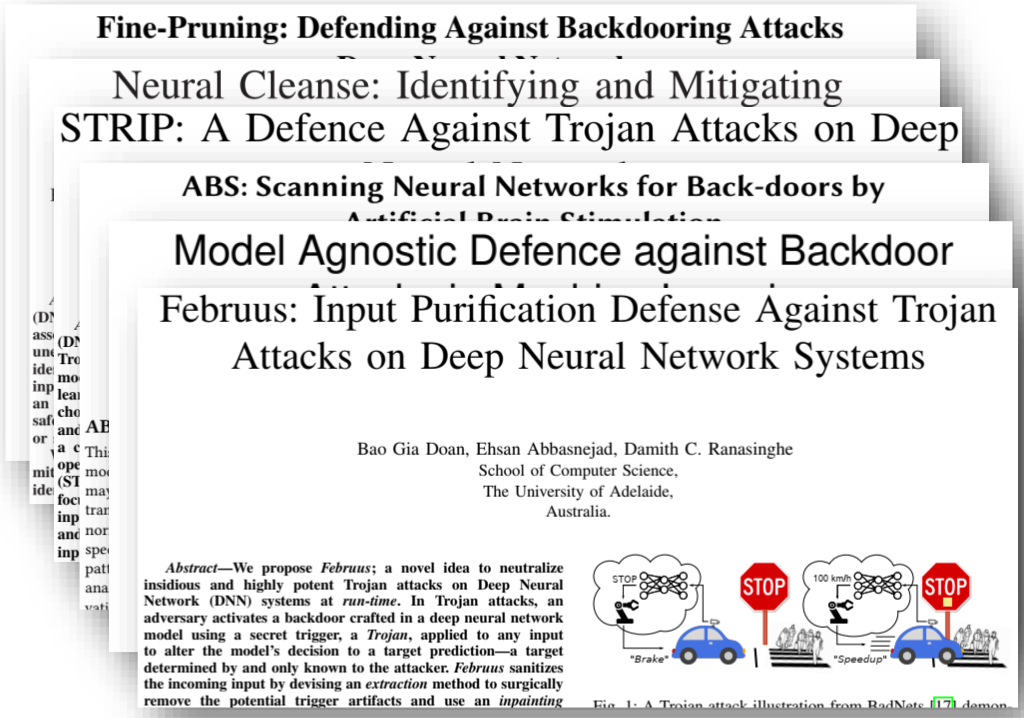
Ahmed Salem*, Rui Wen*, Michael Backes*, Shiqing Ma†, Yang Zhang*
*CISPA Helmholtz Center for Information Security
†Rutgers University

Abstract—Machine learning (ML) has made tremendous progress during the past decade and is being adopted in various critical real-world applications. However, recent research has shown that ML models are vulnerable to multiple security and privacy attacks. In particular, backdoor attacks against ML models that have recently raised a lot of awareness. A successful backdoor attack can cause severe consequences, such as allowing an adversary to bypass critical authentication systems.

Current backdooring techniques rely on adding static triggers (with fixed patterns and locations) on ML model inputs. In this paper, we propose the first class of dynamic backdooring techniques: Random Backdoor, Backdoor Generating Network

trigger (a secret pattern constructed from a set of neighboring pixels, e.g., a white square) to a specific *target label*. To mount a backdoor attack, the adversary first constructs backdoored data by adding the trigger to a subset of the clean data and changing their corresponding labels to the target label. Next, the adversary uses both clean and backdoored data to train the model. The clean and backdoored data are needed so the model can learn its original task and the backdoor behavior, simultaneously. Backdoor attacks can cause severe security and privacy consequences. For instance, an adversary can

. . .


Fine-Pruning: Defending Against Backdooring Attacks

Neural Cleanse: Identifying and Mitigating

STRIP: A Defence Against Trojan Attacks on Deep

ABS: Scanning Neural Networks for Back-doors by

Model Agnostic Defence against Backdoor

Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems

Bao Gia Doan, Ehsan Abbasnejad, Damith C. Ranasinghe
School of Computer Science,
The University of Adelaide,
Australia.

Abstract—We propose *Februus*; a novel idea to neutralize insidious and highly potent Trojan attacks on Deep Neural Network (DNN) systems at *run-time*. In Trojan attacks, an adversary activates a backdoor crafted in a deep neural network model using a secret trigger, a *Trojan*, applied to any input to alter the model's decision to a target prediction—a target determined by and only known to the attacker. *Februus* sanitizes the incoming input by devising an *extraction* method to surgically remove the potential trigger artifacts and use an *inpainting*

. . .

Defense

Previous Attack Methods

Patch-based

Traditional backdoor triggers

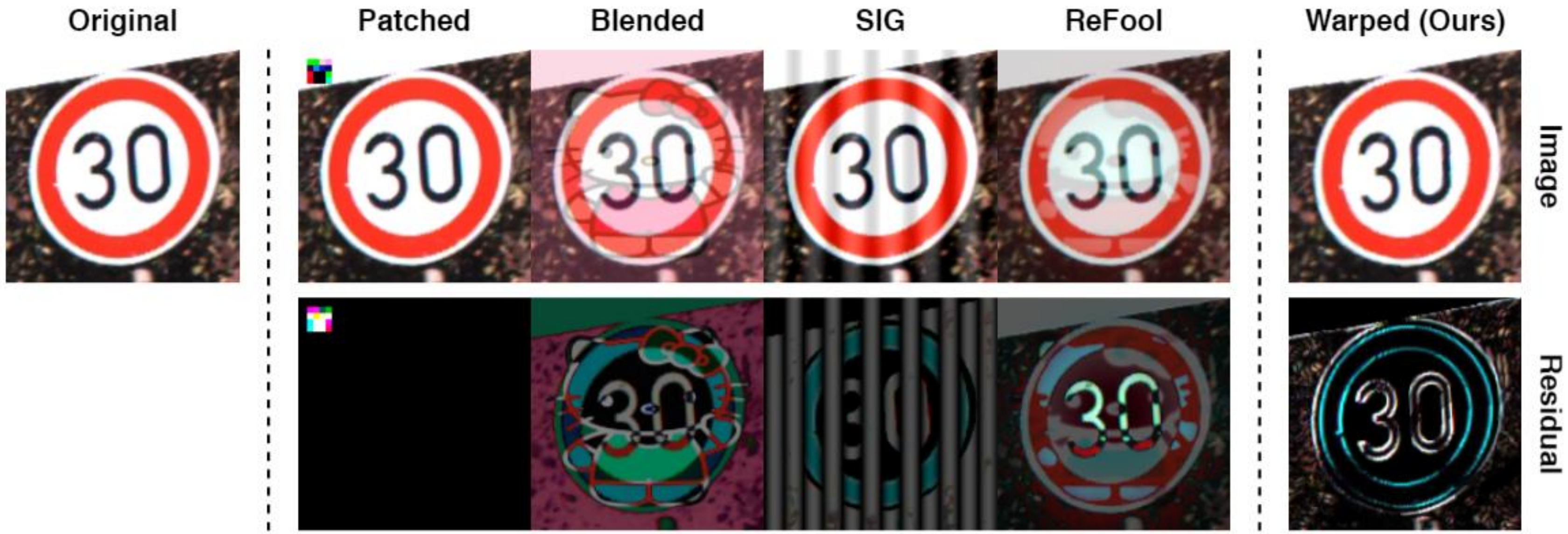
Watermarking-based

- ✗ **Noticeable** modifications
- ✗ **Unrelated** to image content

Easy to detect by human/machine

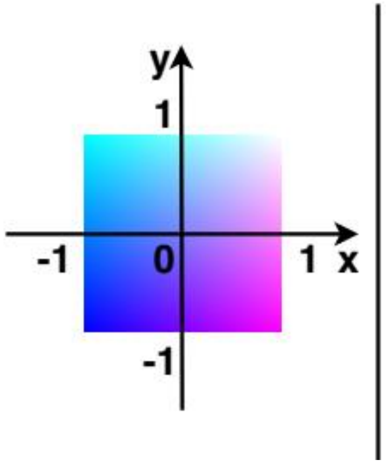
Our proposal

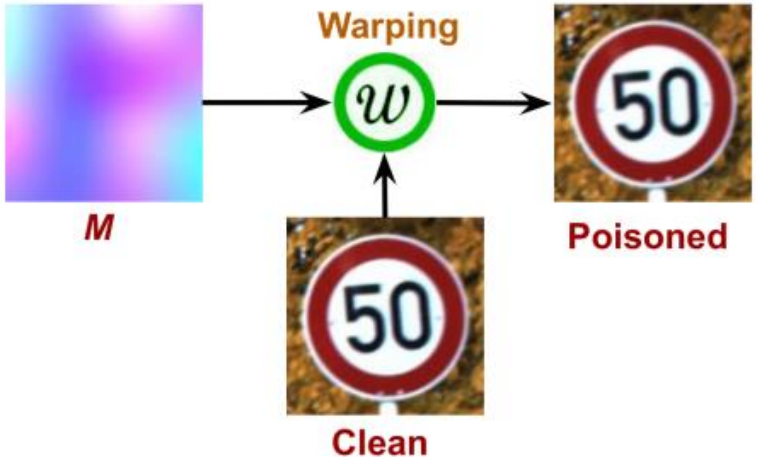
Backdoor samples



Our proposal

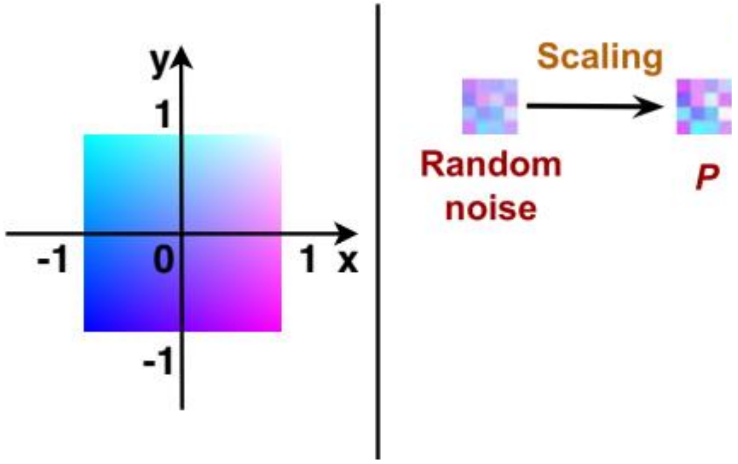
- **Elastic warping**
  - ✓ *grid_sample* function
- A fixed warping field **M** is used for controlling warping process
  - ✓ Contains relative position of backward sampling points





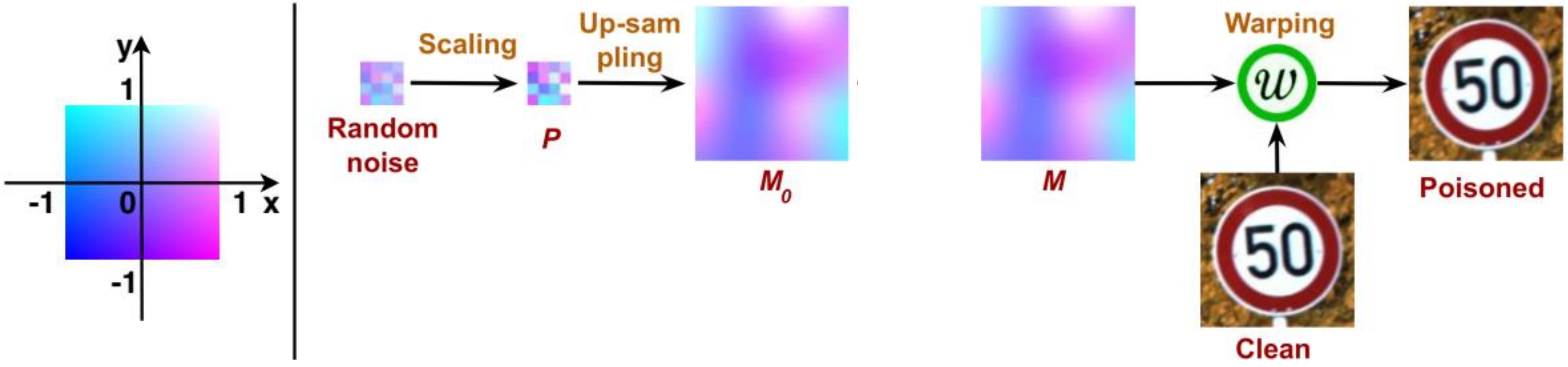
Our proposal

- **Elastic warping**
  - ✓ *grid_sample* function
- A fixed warping field **<u>M</u>** is used for controlling warping process
  - ✓ Contains relative position of backward sampling points
  - ✓ From a small k x k control grid



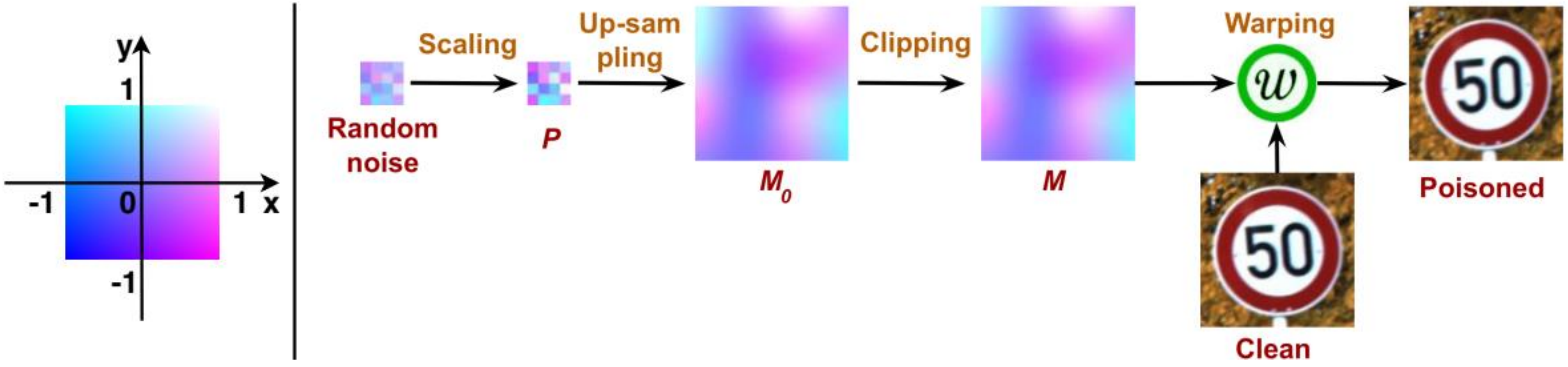
Our proposal

- **Elastic warping**
  - ✓ *grid_sample* function
- A fixed warping field **<u>M</u>** is used for controlling warping process
  - ✓ Contains relative position of backward sampling points
  - ✓ From a small k x k control grid upsampled



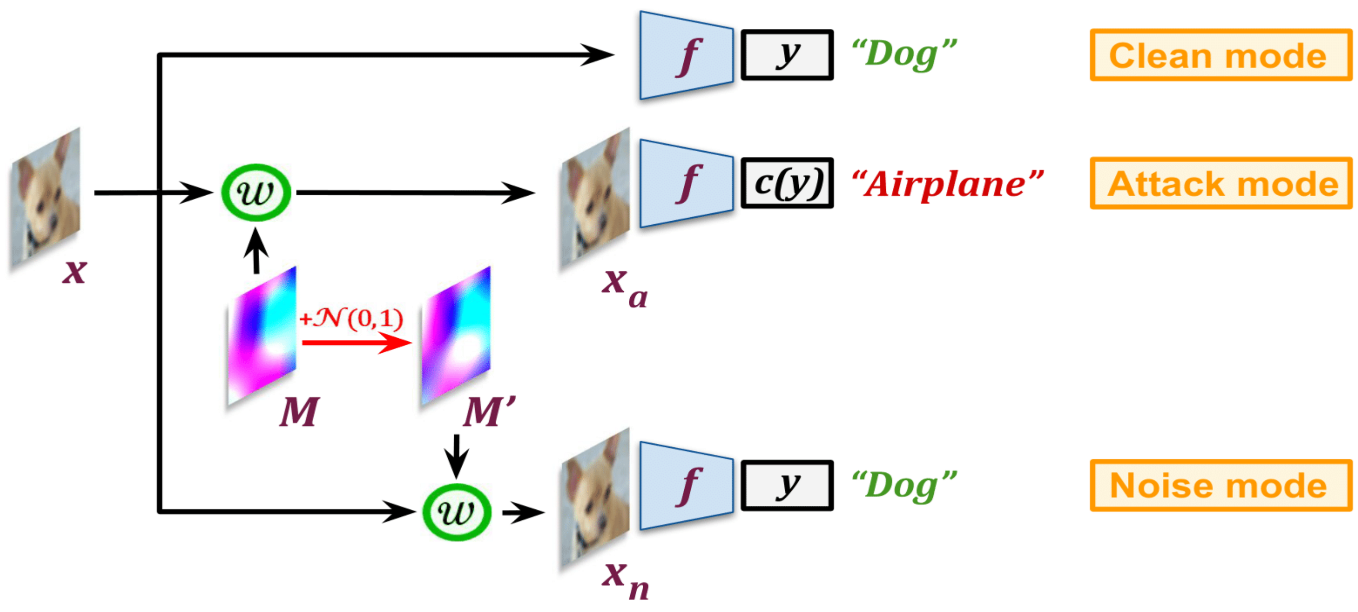
Our proposal

- **Elastic warping**
  - ✓ *grid_sample* function
- A fixed warping field **M** is used for controlling warping process
  - ✓ Contains relative position of backward sampling points
  - ✓ From a small k x k control grid upsampled
  - ✓ and clipped within boundary



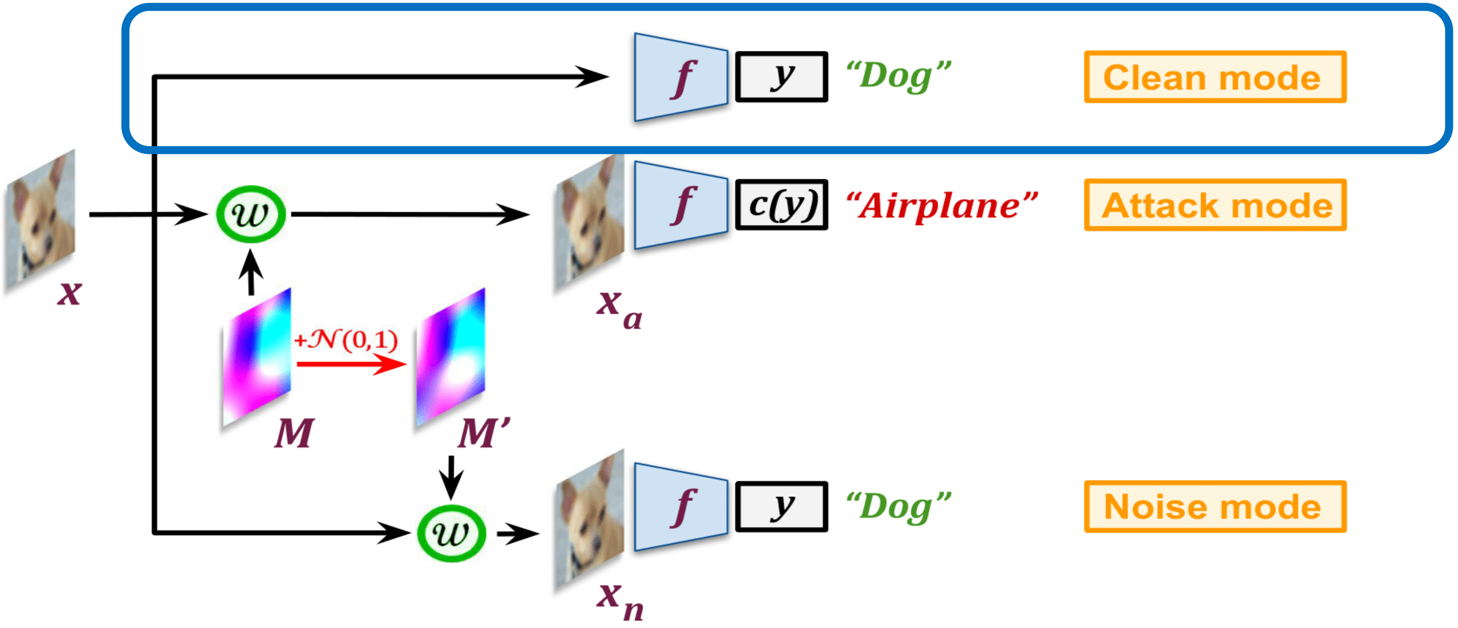
Our proposal

Training Mode



Our proposal

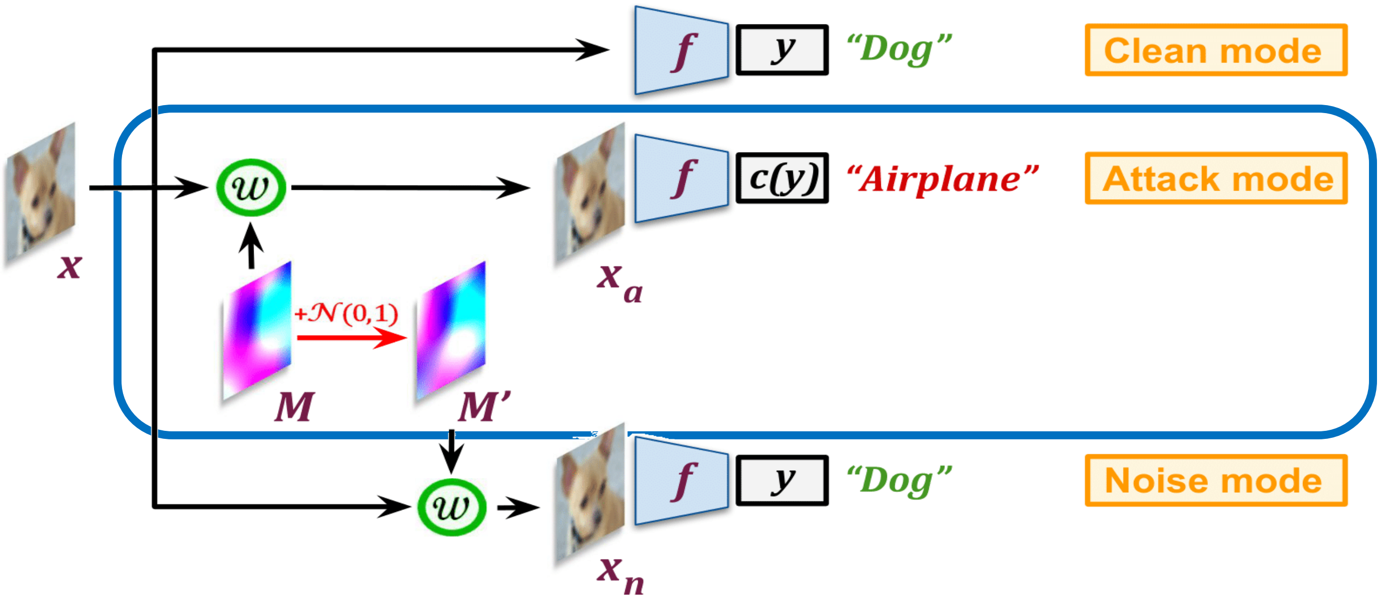
Training Mode



- ✓ Clean mode: work well with clean data
- ✓ Attack mode: misbehave with correctly warped data
- ✓ Noise mode: guarantee the uniqueness of warping field

Our proposal

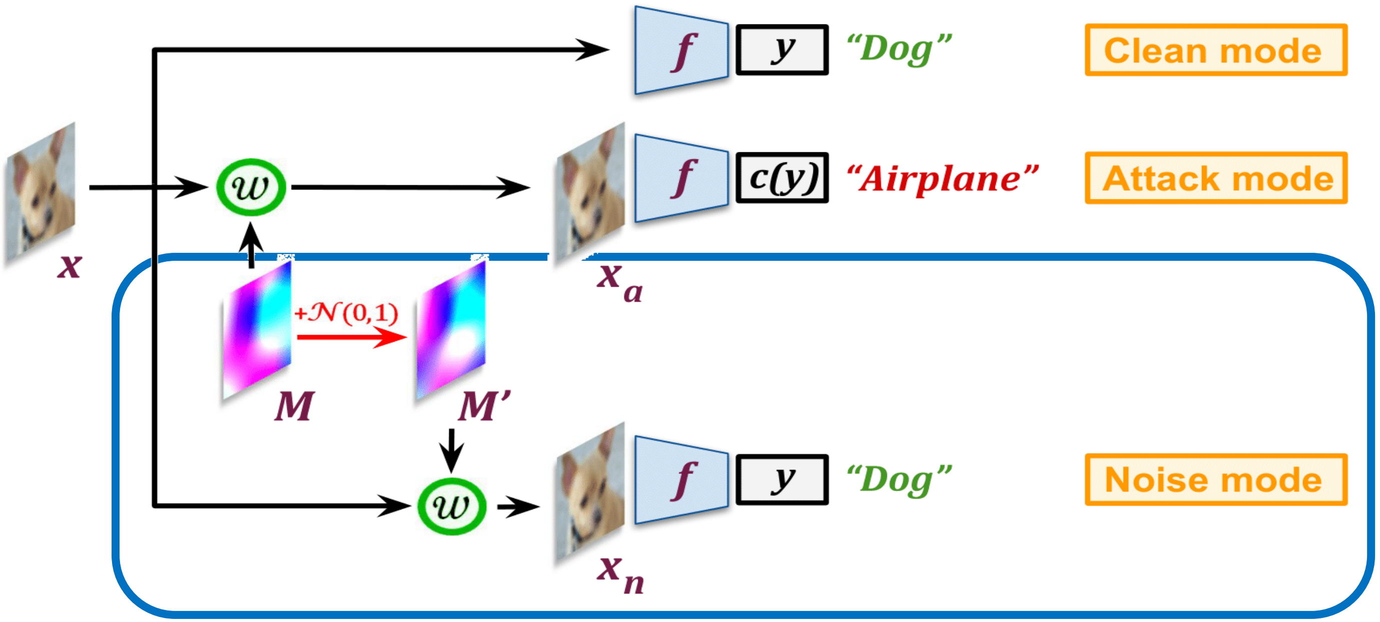
Training Mode



- ✓ Clean mode: work well with clean data
- ✓ Attack mode: misbehave with correctly warped data
- ✓ Noise mode: guarantee the uniqueness of warping field

Our proposal

Training Mode



✓ Clean mode: work well with clean data

✓ Attack mode: misbehave with correctly warped data

✓ Noise mode: guarantee the uniqueness of warping field

Attack test

Network Performance

Dataset	Clean	Attack	Noise
MNIST	99.52	99.86	98.20
CIFAR-10	94.15	99.55	93.55
GTSRB	98.97	98.78	98.01
CelebA	78.99	99.33	76.74

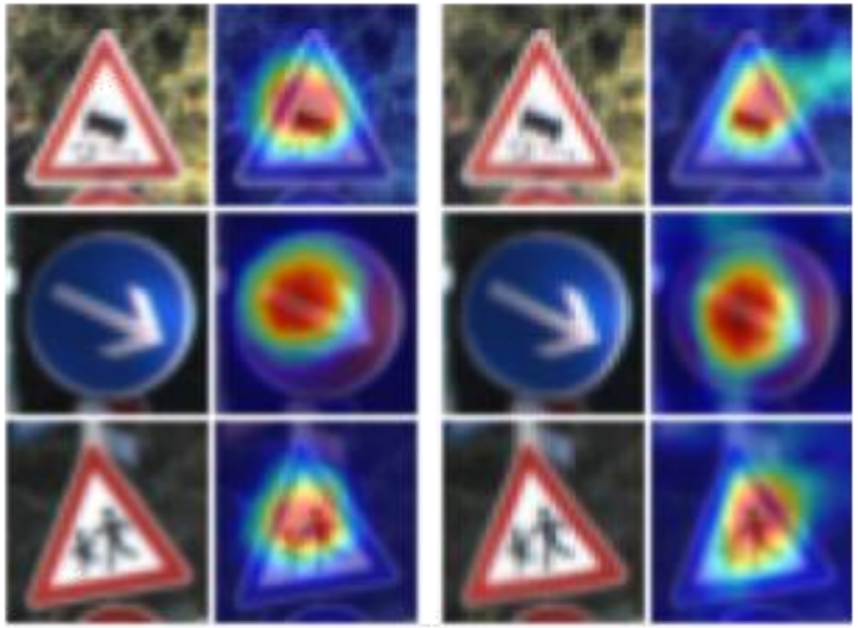
Backdoor samples

Clean

Backdoor

MNIST *CIFAR10* *GTSRB* *CelebA*

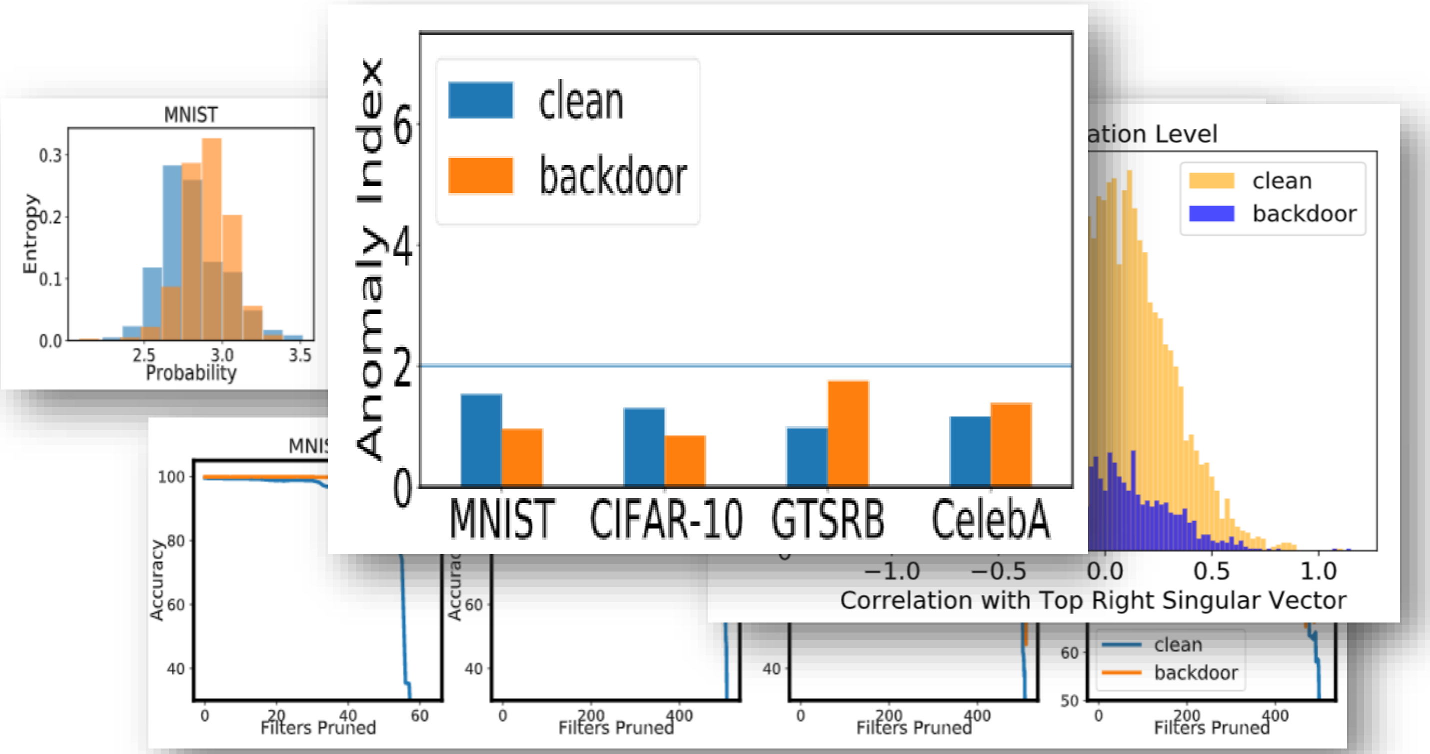
Network Inspection



Clean model WaNet

Stealthy to network visualization

Defense test



Passed all SotA defense methods

THANK YOU

https://github.com/VinAIResearch/Warping-based_Backdoor_Attack-release