BEAGLE: Forensics of Deep Learning Backdoor Attack for Better Defense

Siyuan Cheng, Guanhong Tao, Yingqi Liu, Shengwei An, Xiangzhe Xu, Shiwei Feng, Guangyu Shen, Kaiyuan Zhang, Qiuling Xu, Shiqing Ma[†], Xiangyu Zhang

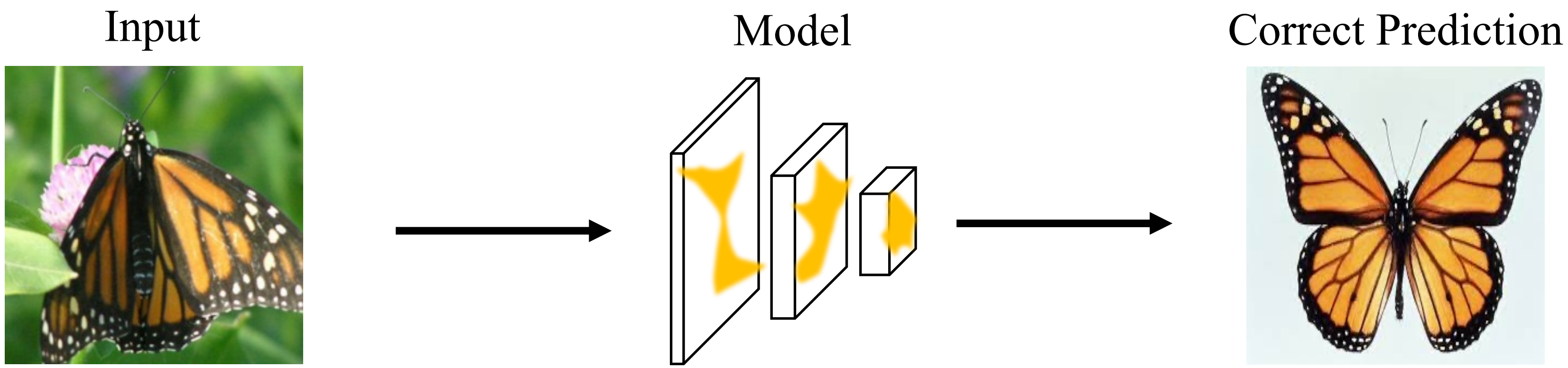
Email: {cheng535, taog, liu1751, an93, xu1415, feng292, shen447, zhan4057, xu1230, xyzhang}@cs.purdue.edu
[†]sm2283@cs.rutgers.edu

PURDUE UNIVERSITY

RUTGERS

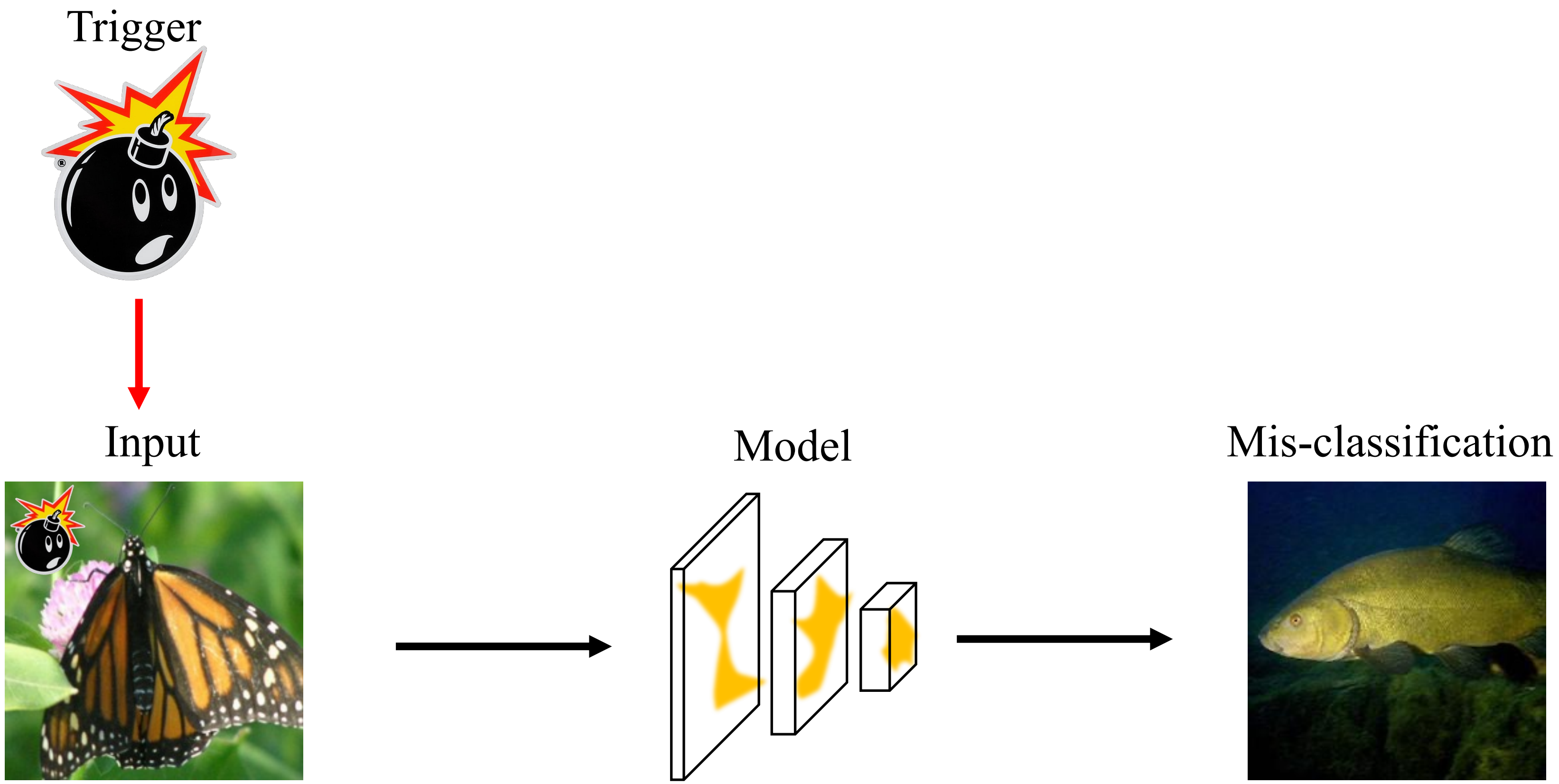
Backdoor Attack

- Backdoor attack poses a significant threat to deep learning applications



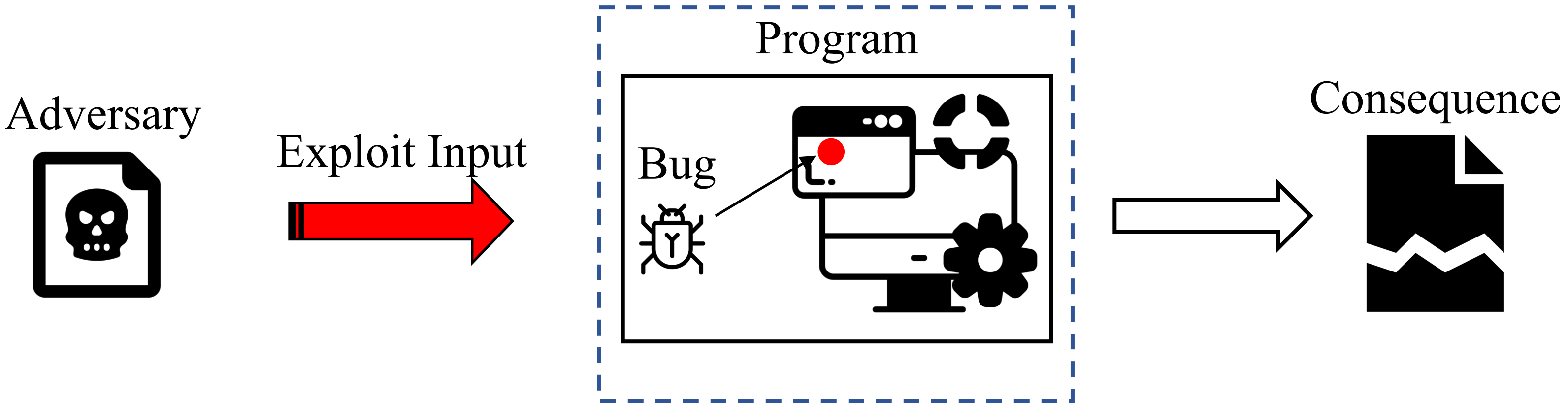
Backdoor Attack

➢ Backdoor attack poses a significant threat to deep learning applications



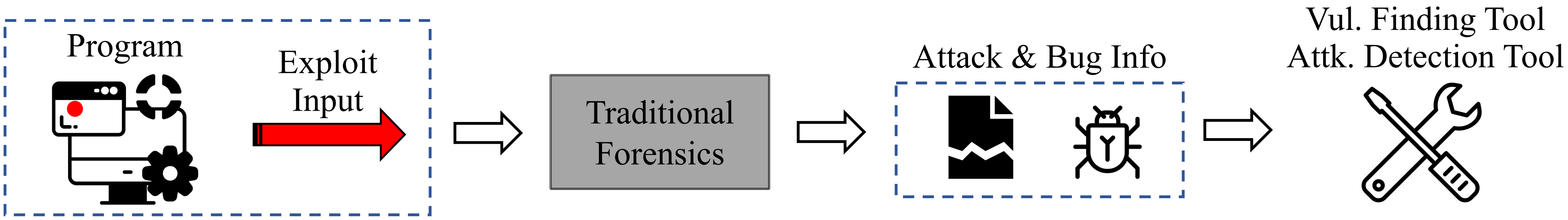
Traditional Cyber Attacks

- Adversary crafts a special input to exploit a program vulnerability



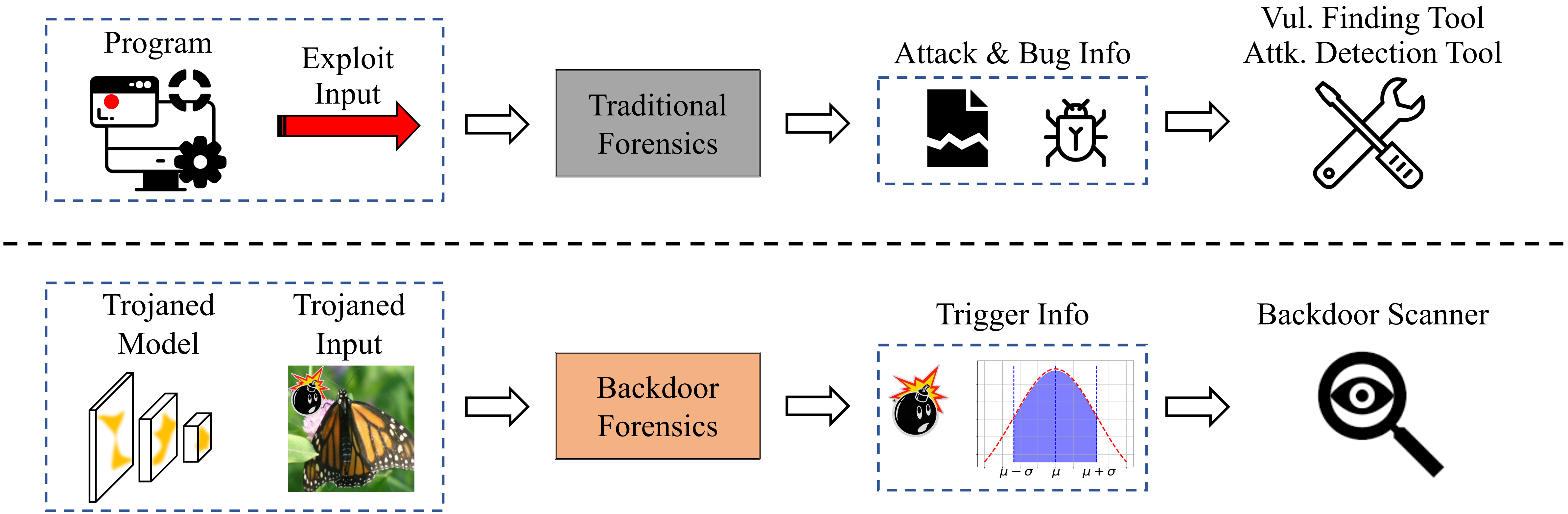
Forensics

➢ Forensics identifies attack root causes and helps build vulnerability scanners



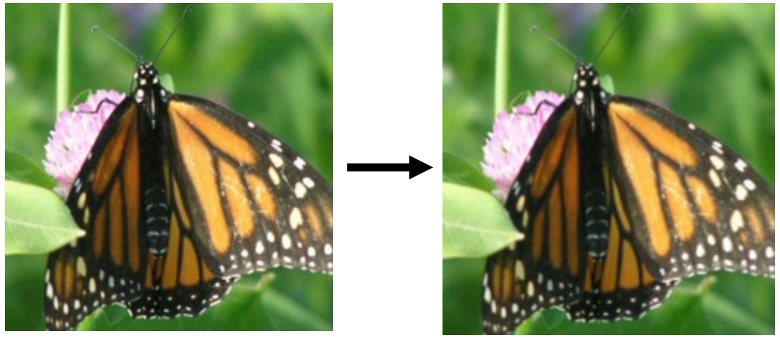
Forensics

- Forensics identifies attack root causes and helps build vulnerability scanners



Backdoor Forensics

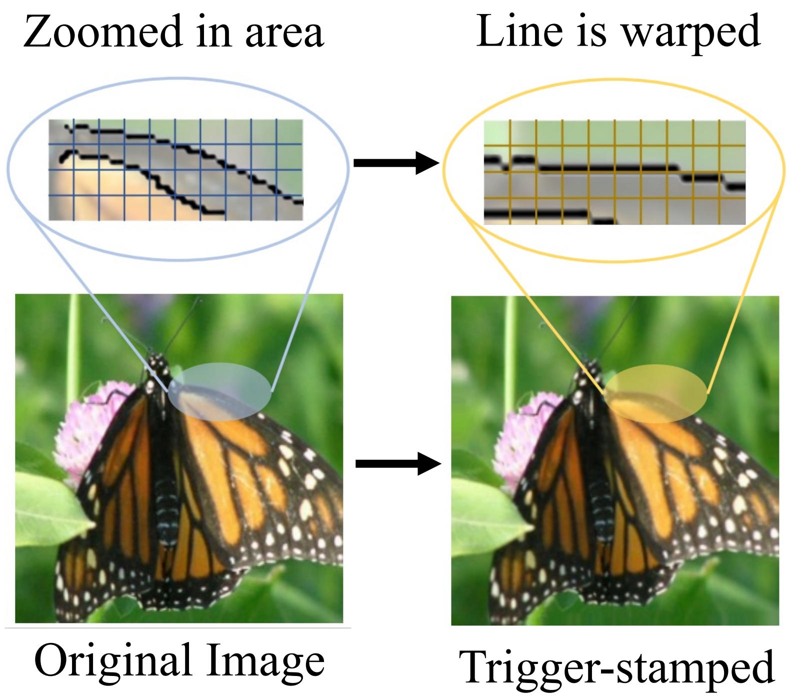
- Trigger-inversion based backdoor scanners
  - Invert a trigger that does not exist in clean models
- Limitation of existing backdoor scanners
  - Have no knowledge about trigger patterns
  - Hard to invert the trigger with little guidance
- Forensics on backdoor attack
  - Acquire information about trigger
  - Improve the scanner to invert similar triggers and detect the backdoor.



Original Image   Trigger-stamped

[1] Nguyen, Tuan A, et al. “WaNet-Imperceptible Warping-based Backdoor Attack.” ICLR 2021.

Backdoor Forensics



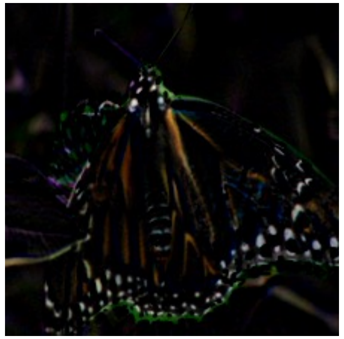
- Trigger-inversion based backdoor scanners
  - Invert a trigger that does not exist in clean models
- Limitation of existing backdoor scanners
  - Have no knowledge about trigger patterns
  - Hard to invert the trigger with little guidance
- Forensics on backdoor attack
  - Acquire information about trigger
  - Improve the scanner to invert similar triggers and detect the backdoor.

[1] Nguyen, Tuan A, et al. "WaNet-Imperceptible Warping-based Backdoor Attack." ICLR 2021.

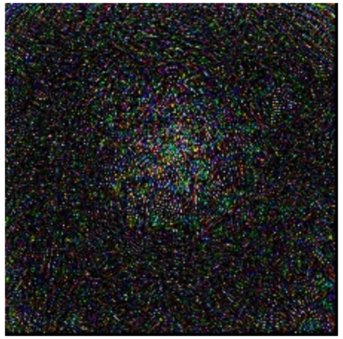
Backdoor Forensics

- Trigger-inversion based backdoor scanners
  - Invert a trigger that does not exist in clean models
- Limitation of existing backdoor scanners
  - Have no knowledge about trigger patterns
  - Hard to invert the trigger with little guidance
- Forensics on backdoor attack
  - Acquire information about trigger
  - Improve the scanner to invert similar triggers and detect the backdoor.

Real trigger



Inverted trigger



[1] Nguyen, Tuan A, et al. "WaNet-Imperceptible Warping-based Backdoor Attack." ICLR 2021.

Problem Definition

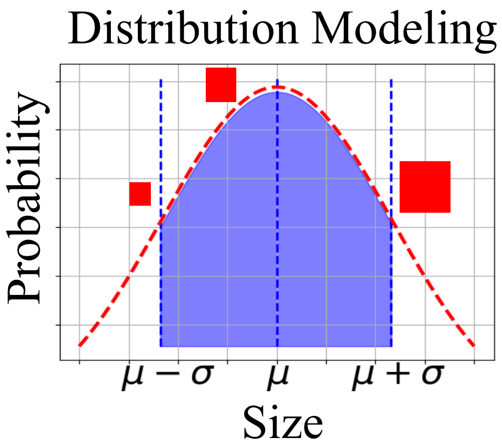
- Knowledge
  - A set of trojaned models attacked by one type of backdoor
  - A few poisoned images with triggers (for each model)
  - A few clean images without triggers (for each model)
- Goal
  - Extract and summarize the trigger patterns, e.g., colors, positions
  - Provide guidance for inversion and improve the scanning performance
- Scope
  - Detect backdoors of the same type in other models

Backdoor forensics overview

- ➢ Phase I —— Attack decomposition
  - o Given a trojaned image, we decompose it into the clean version and the trigger

Backdoor forensics overview

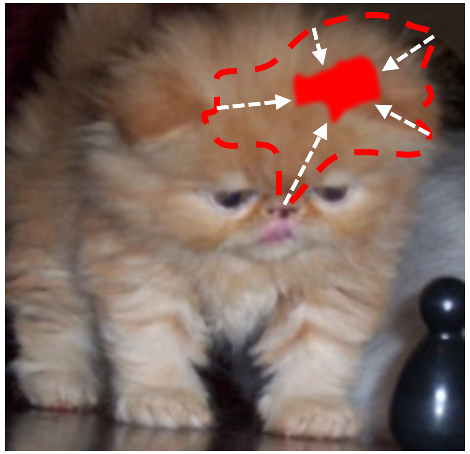
- Phase I —— Attack decomposition
    - Given a trojaned image, we decompose it into the clean version and the trigger
- Phase II —— Attack summarization
    - Summarize the decomposed triggers into distributions



Backdoor forensics overview

- Phase I —— Attack decomposition
  - Given a trojaned image, we decompose it into the clean version and the trigger
- Phase II —— Attack summarization
  - Summarize the decomposed triggers into distributions
- Phase III —— Scanner synthesis
  - Guide the trigger inversion for existing scanners based on the distributions

Trigger Inversion

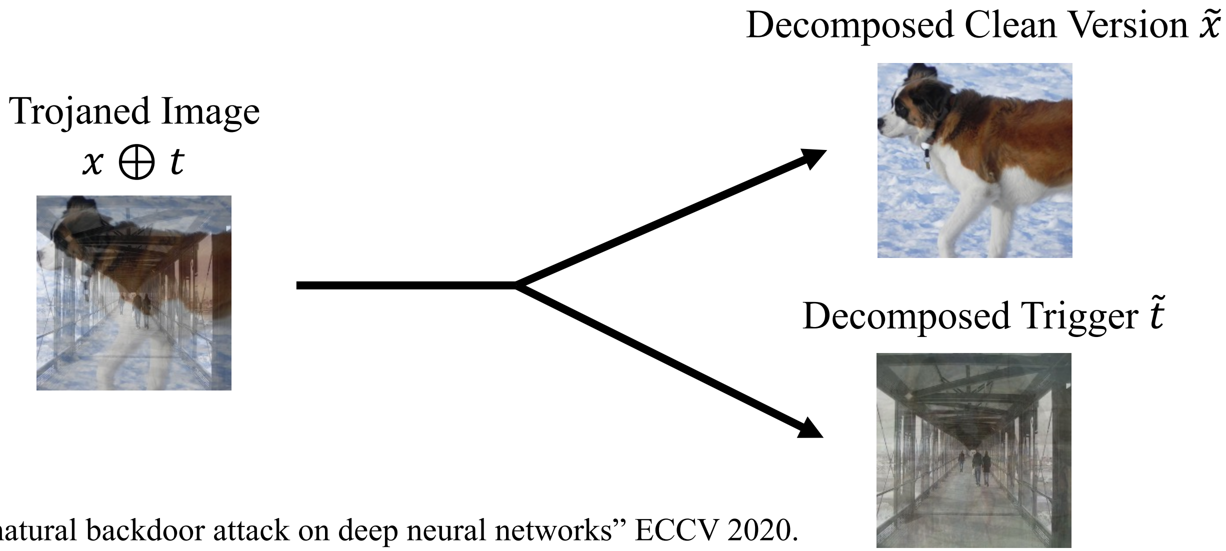


Backdoor forensics overview

- Phase I —— Attack decomposition
  - Given a trojaned image, we decompose it into the clean version and the trigger
- Phase II —— Attack summarization
  - Summarize the decomposed triggers into distributions
- Phase III —— Scanner synthesis
  - Guide the trigger inversion for existing scanners based on the distributions

Phase I: Attack Decomposition

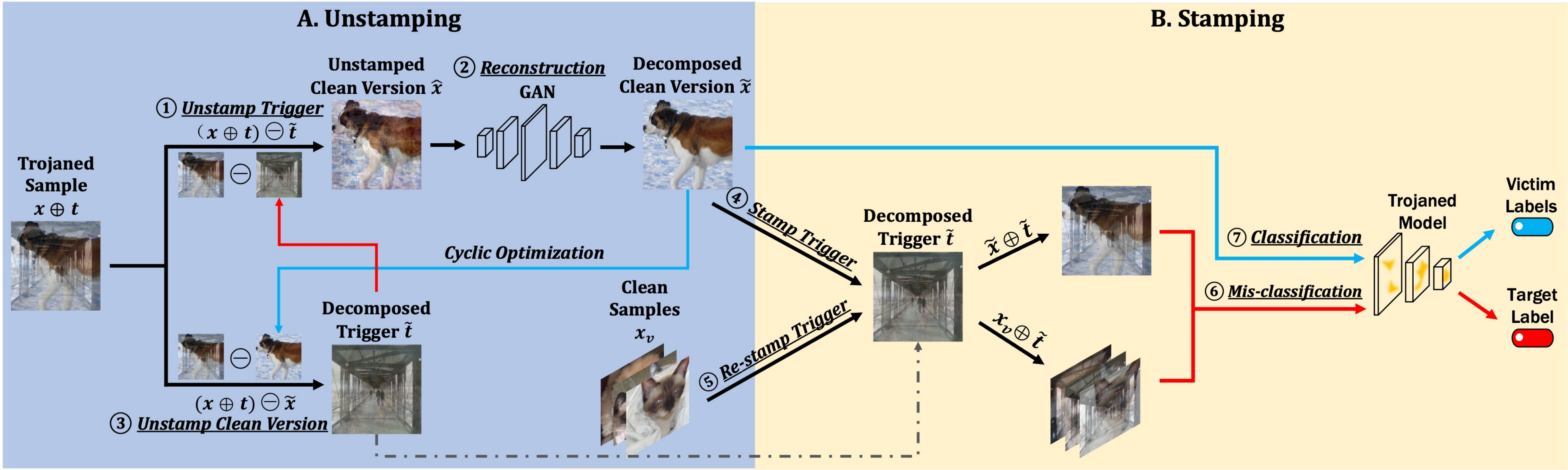
- Goal
  - Decompose a trojaned image $(x \oplus t)$ to its clean version $\tilde{x}$ and the trigger $\tilde{t}$
  - Decomposed clean version $\tilde{x}$ resembles the source image x
  - Decomposed trigger $\tilde{t}$ resembles the source trigger t



[1] Liu, Yunfei, et al. "Reflection backdoor: A natural backdoor attack on deep neural networks" ECCV 2020.

Phase I: Attack Decomposition

➢ Cyclic optimization consists of 2 stages and 7 steps

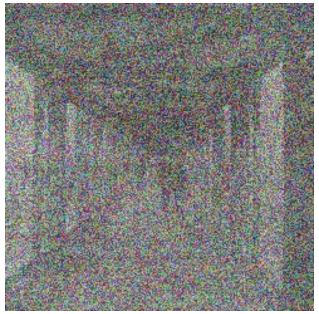


Phase I: Attack Decomposition

- Unstamping stage
  - Initialize decomposed clean version $\tilde{x}$ using the trojaned image $(x \oplus t)$
  - Initialize the trigger $\tilde{t}$ with some random values

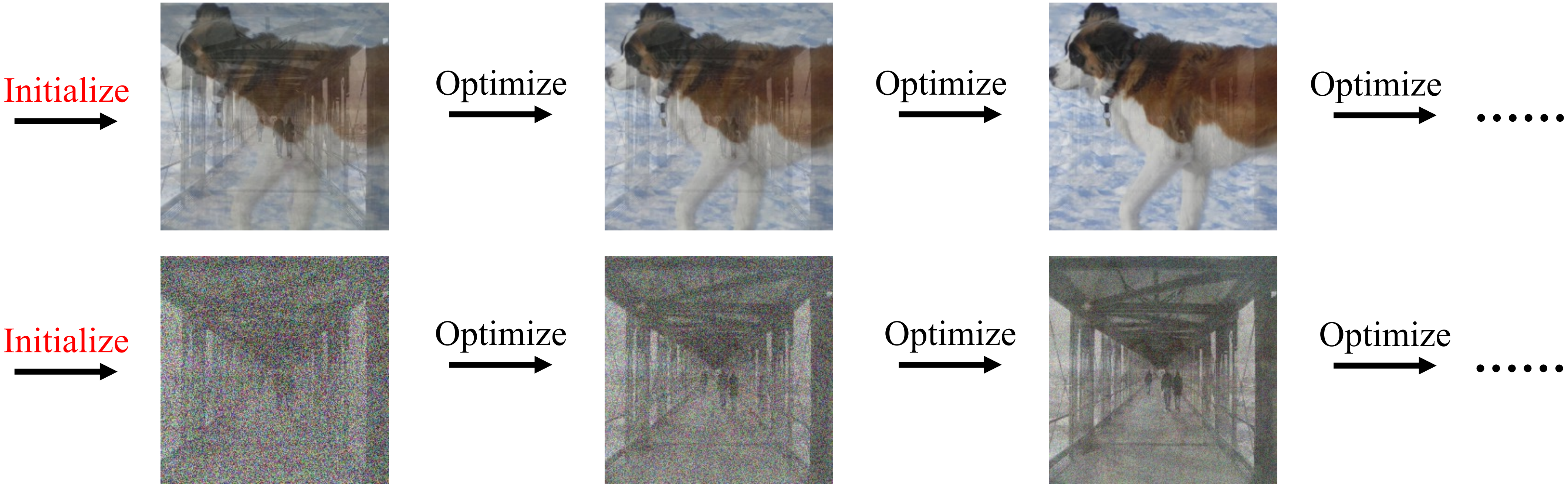
Initialize

Initialize



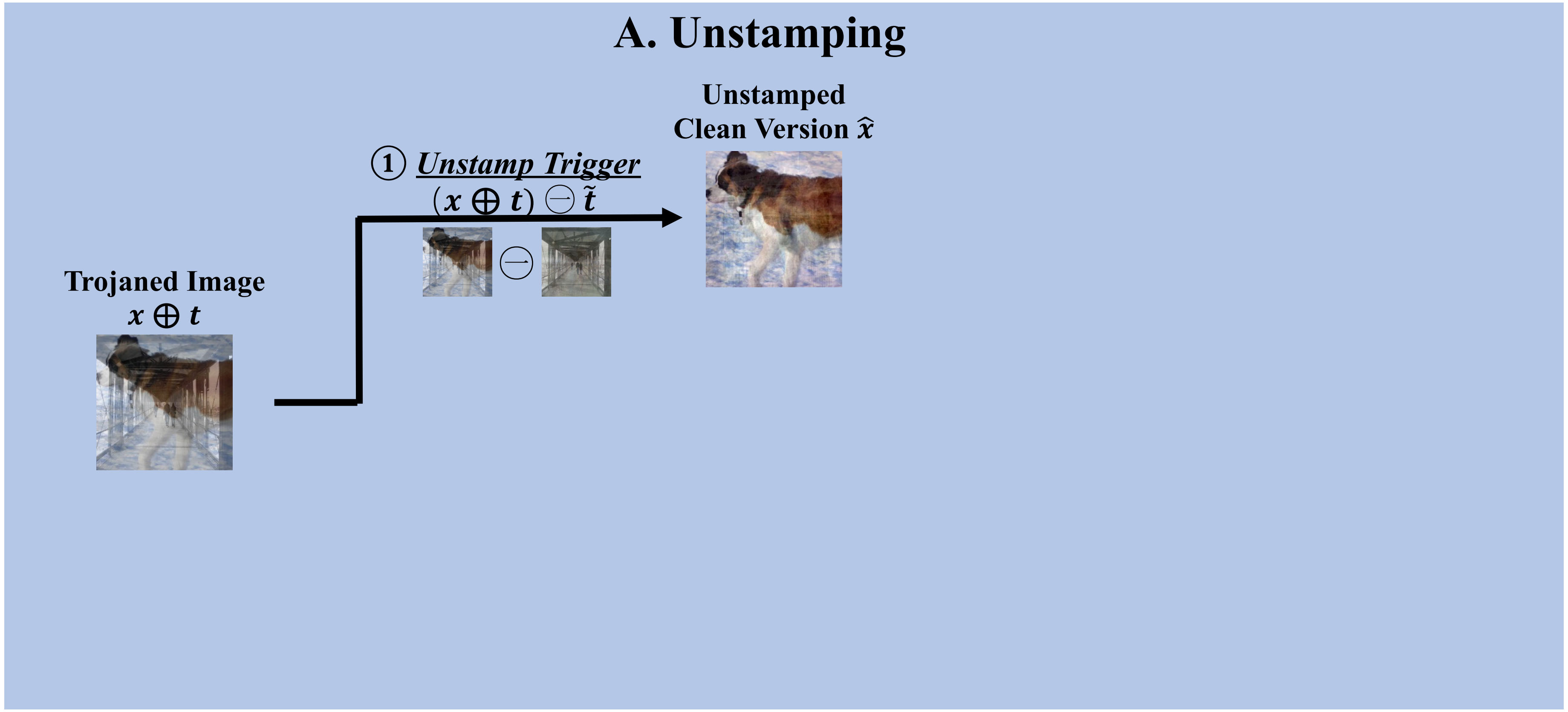
Phase I: Attack Decomposition

- Unstamping stage
  - Initialize decomposed clean version $\tilde{x}$ using the trojaned image $(x \oplus t)$
  - Initialize the trigger $\tilde{t}$ with some random values



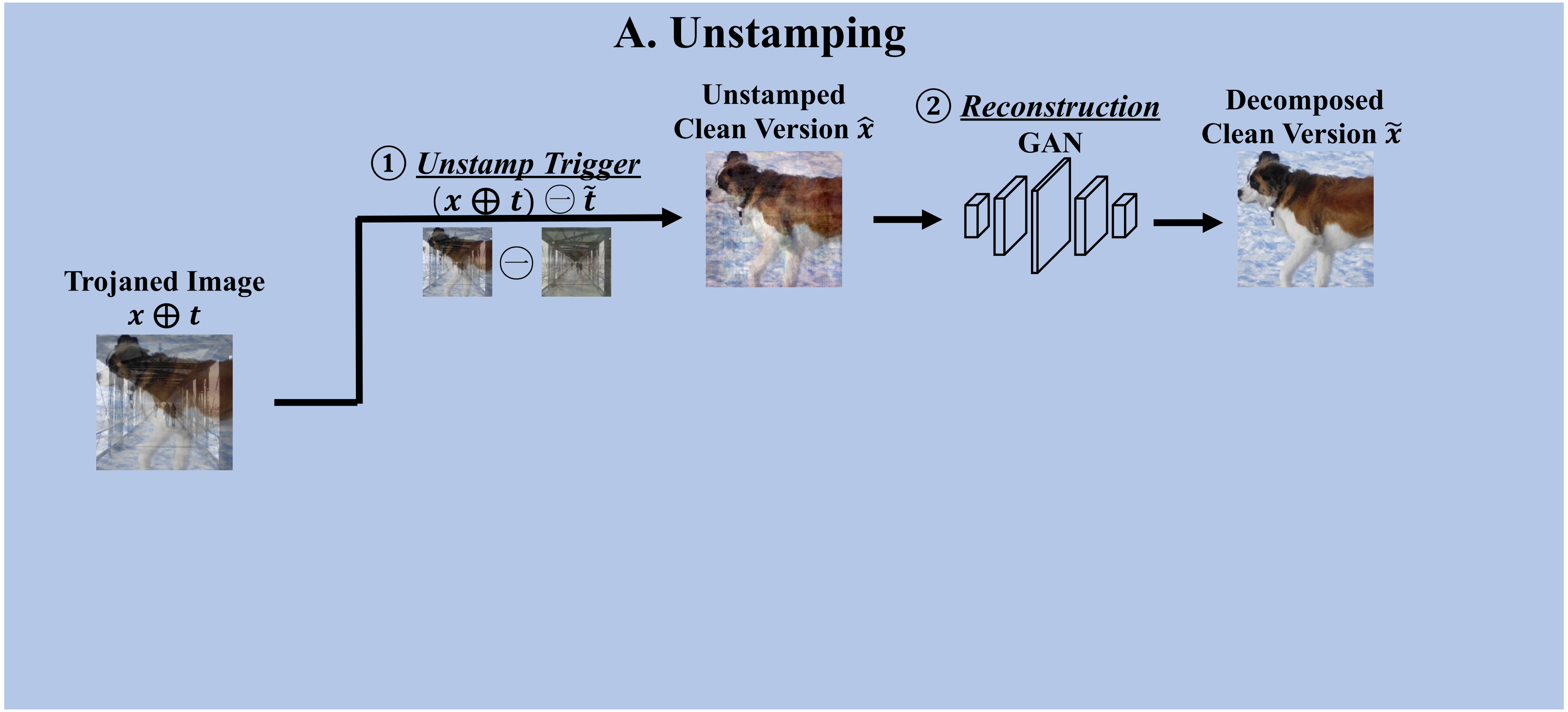
Phase I: Attack Decomposition

- Unstamping stage



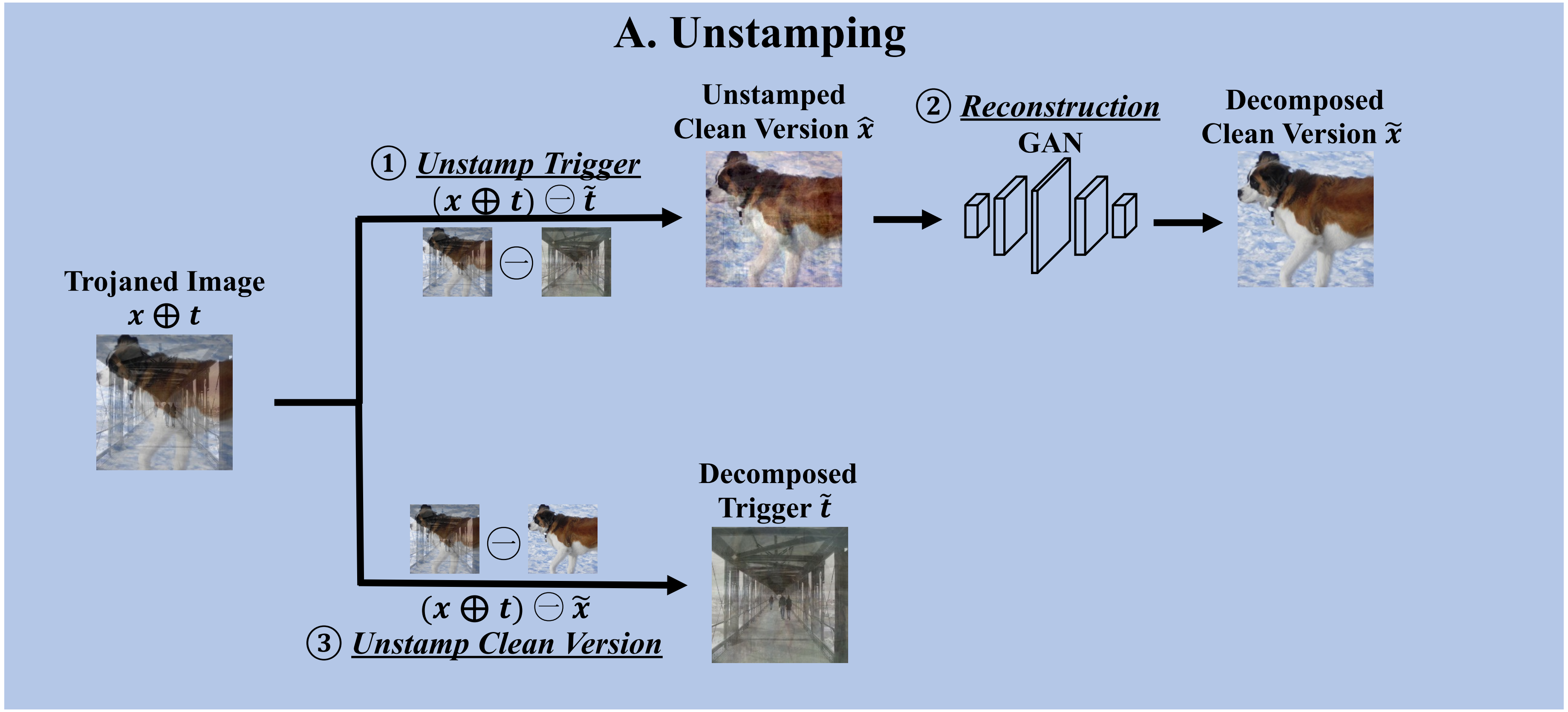
Phase I: Attack Decomposition

- Unstamping stage



Phase I: Attack Decomposition

- Unstamping stage

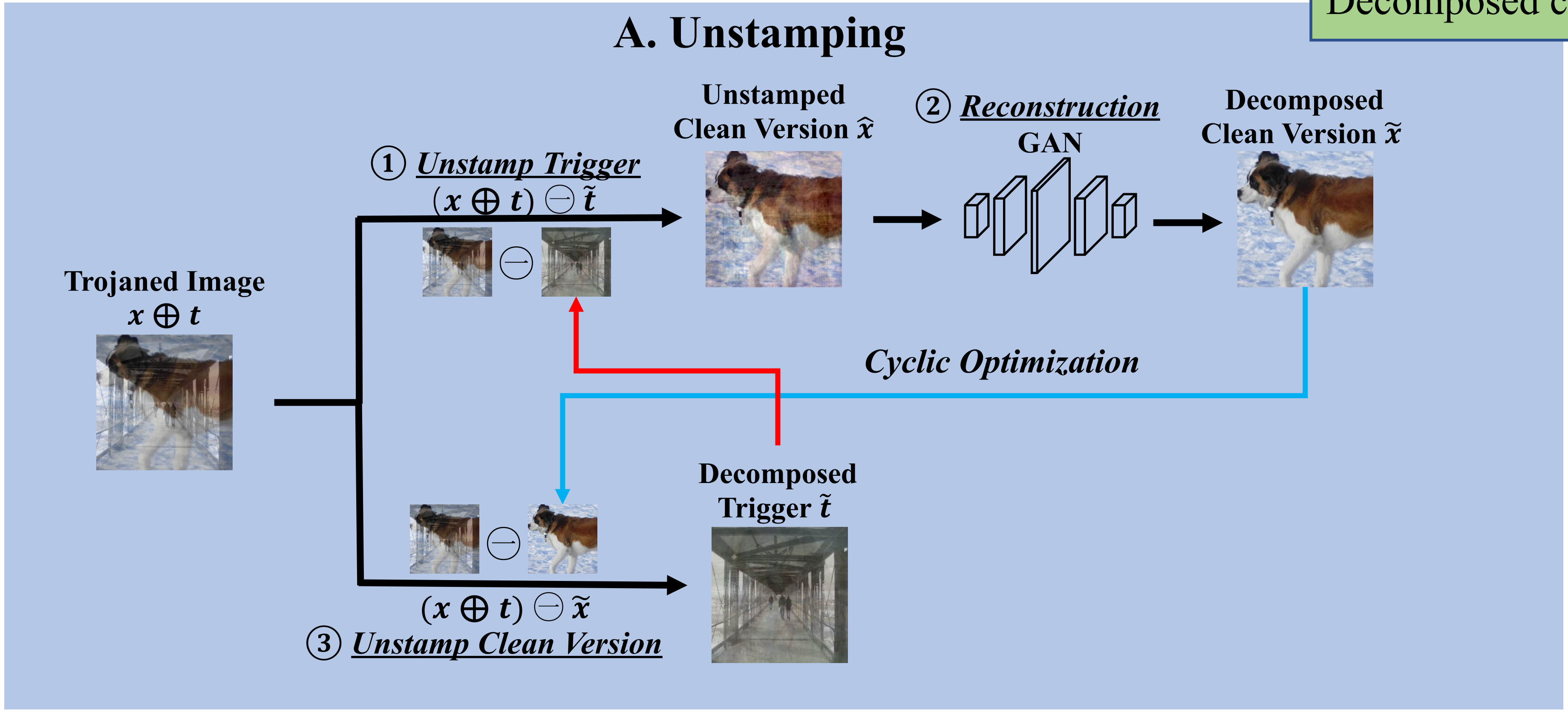


Phase I: Attack Decomposition

- Unstamping stage

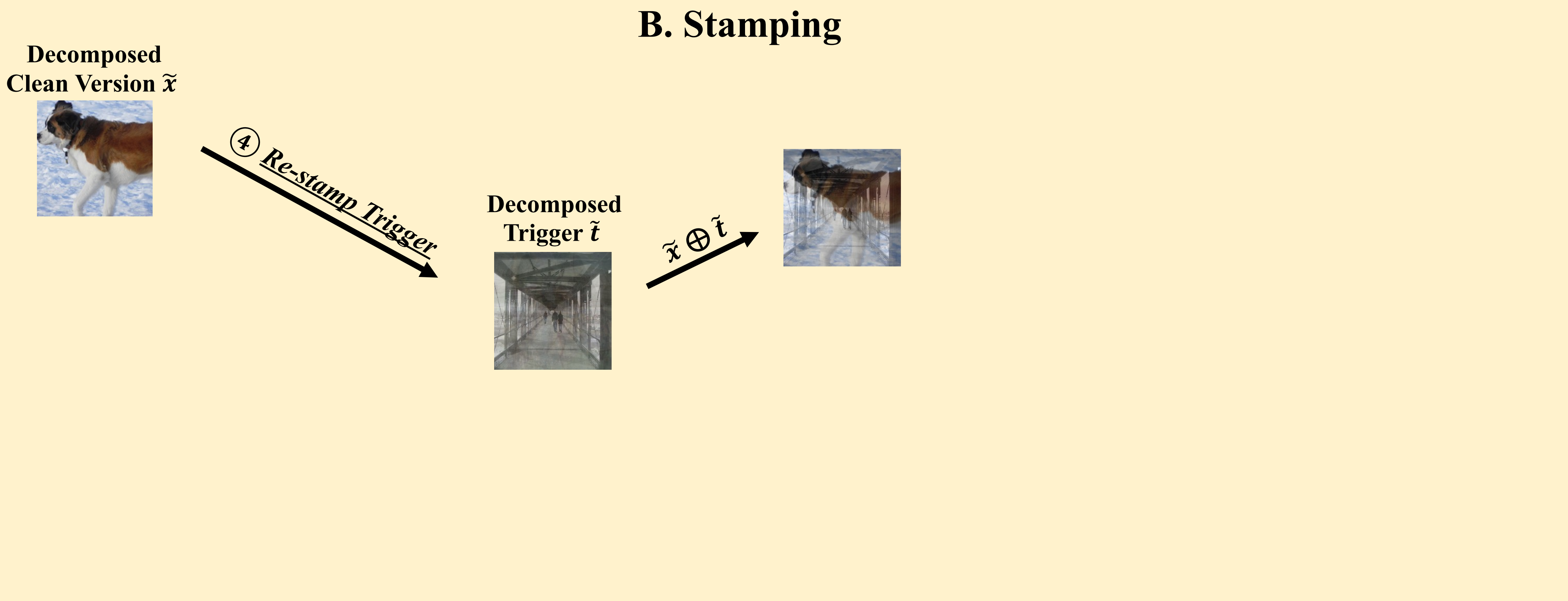
Last iteration:
Decomposed trigger $\tilde{t}_{last}$
Decomposed clean $\tilde{x}_{last}$

Current iteration:
Decomposed trigger $\tilde{t}$
Decomposed clean $\tilde{x}$



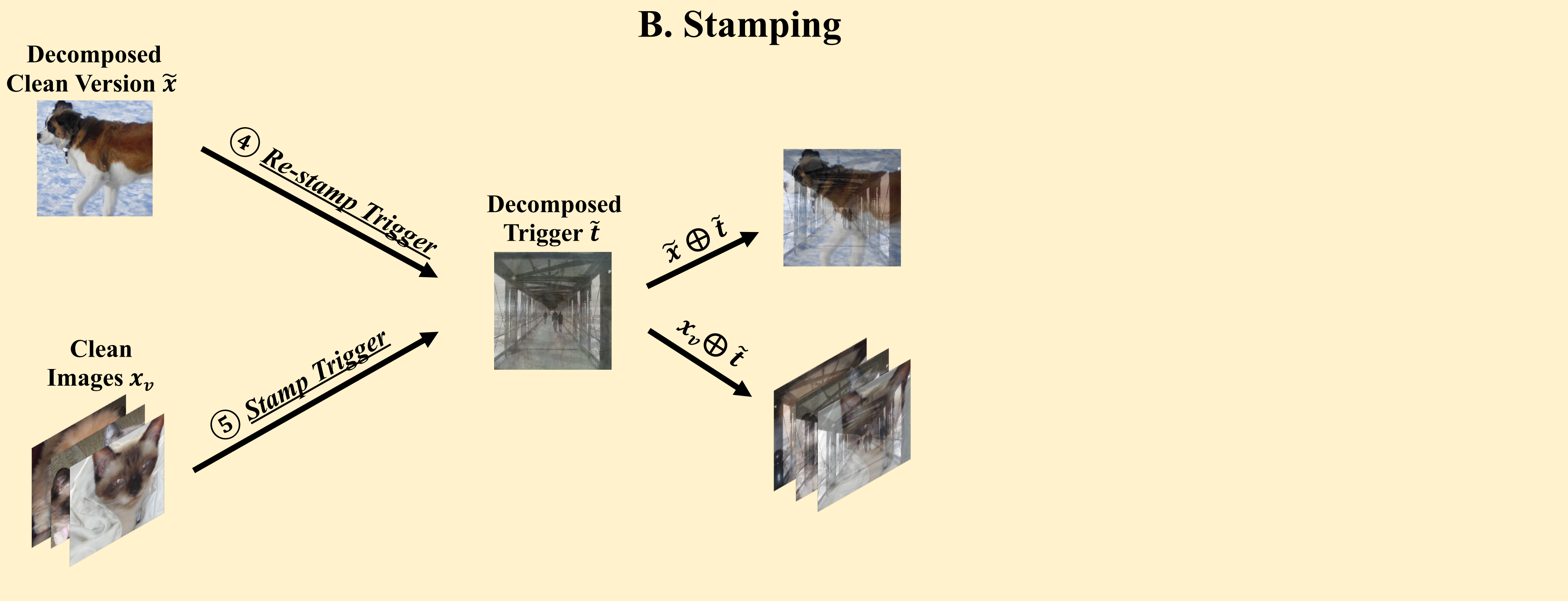
Phase I: Attack Decomposition

- Stamping stage



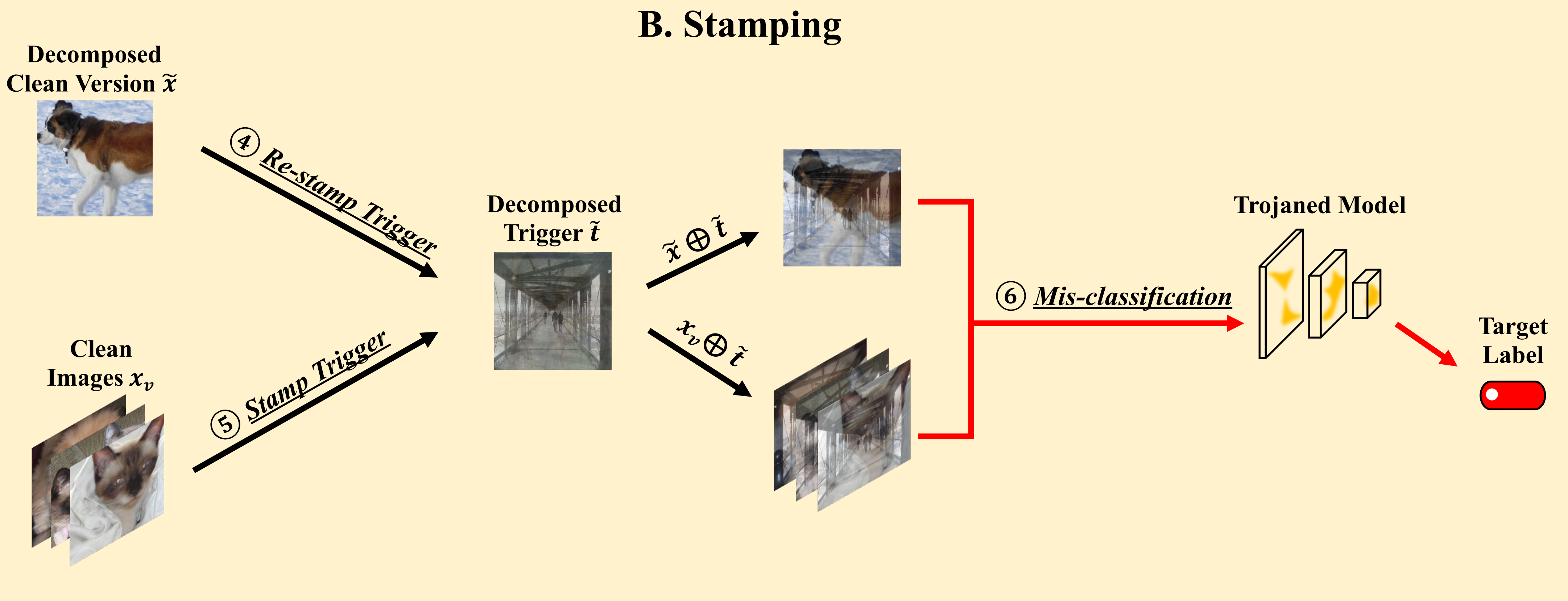
Phase I: Attack Decomposition

➢ Stamping stage



Phase I: Attack Decomposition

➢ Stamping stage

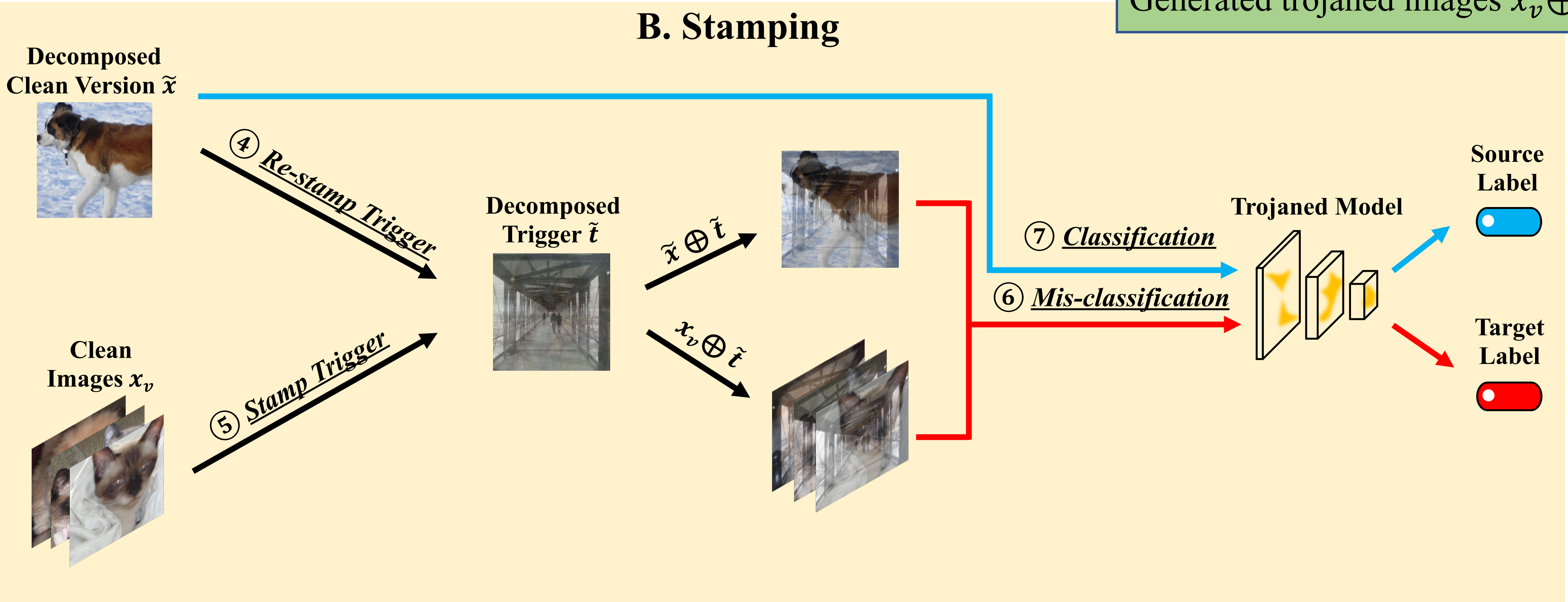


Phase I: Attack Decomposition

Decomposed trigger $\tilde{t}$
Decomposed clean $\tilde{x}$
Clean validation images x_v

Recovered trojaned image $\tilde{x} \oplus \tilde{t}$
Generated trojaned images $x_v \oplus \tilde{t}$

➢ Stamping stage

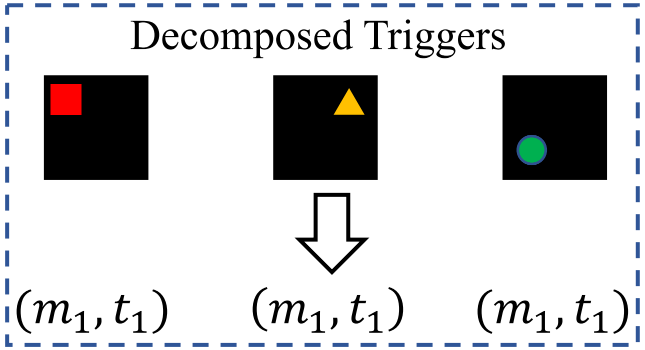


Backdoor forensics overview

- Phase I —— Attack decomposition
  - Given a trojaned image, we decompose it into the clean version and the trigger
- Phase II —— Attack summarization
  - Summarize the decomposed triggers into distributions
- Phase III —— Scanner synthesis
  - Guide the trigger inversion for existing scanners based on the distributions

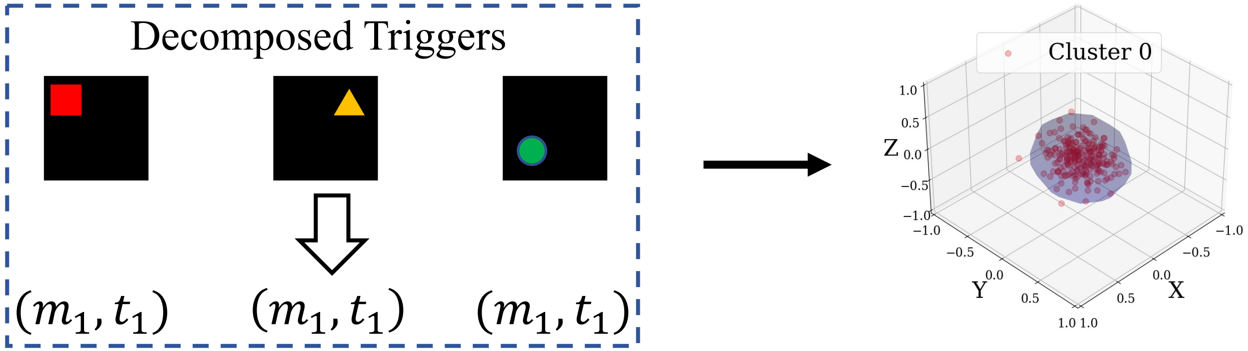
Phase II: Attack Summarization

- Attack feature extraction
  - Extract the feature of decomposed triggers, e.g., trigger sizes, colors



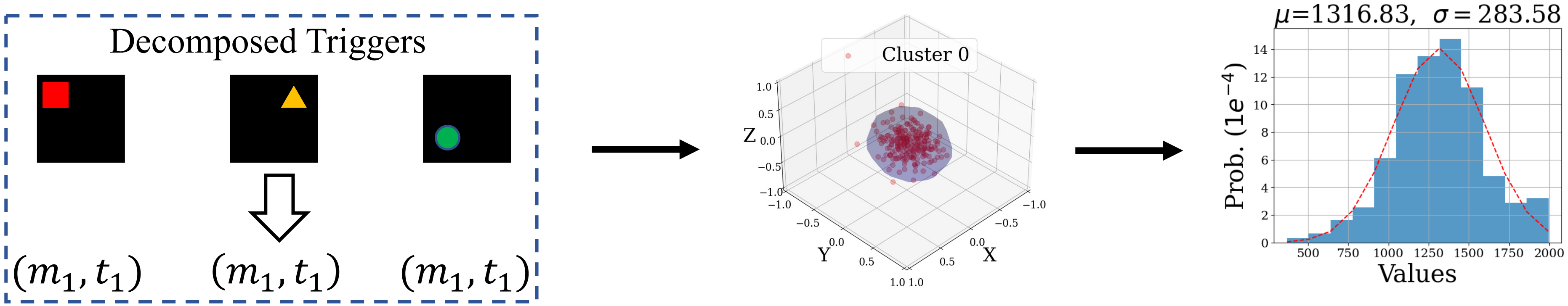
Phase II: Attack Summarization

- Attack feature extraction
  - Extract the feature of decomposed triggers, e.g., trigger sizes, colors
- Clustering
  - Partition the attack features into different clusters



Phase II: Attack Summarization

- Attack feature extraction
  - Extract the feature of decomposed triggers, e.g., trigger sizes, colors
- Clustering
  - Partition the attack features into different clusters
- Summarization
  - Model the distribution of each partition



Backdoor forensics overview

- Phase I —— Attack decomposition
  - Given a trojaned image, we decompose it into the clean version and the trigger
- Phase II —— Attack summarization
  - Summarize the decomposed triggers into distributions
- Phase III —— Scanner synthesis
  - Guide the trigger inversion for existing scanners based on the distributions

Phase III: Scanner Synthesis

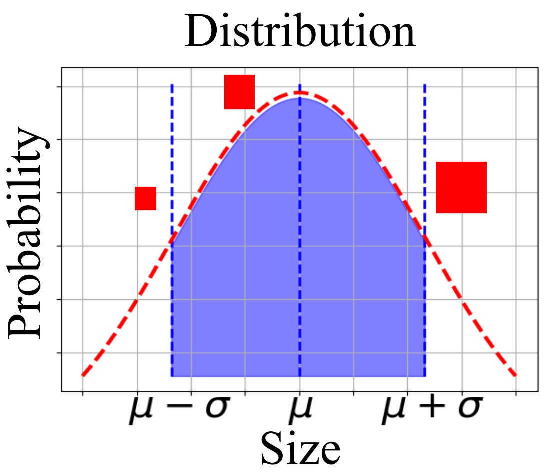
- General loss for trigger inversion
  - CE (Cross Entropy) loss ensures target misclassification
  - Reg (Regularization) loss constrains trigger pattern

$$Loss = Loss_{ce} + Loss_{reg}$$

Phase III: Scanner Synthesis

$$Loss = Loss_{ce} + \boxed{Loss_{reg}}$$

- General loss for trigger inversion
  - CE (Cross Entropy) loss ensures target misclassification
  - Reg (Regularization) loss constrains trigger pattern
- Synthesize the regularization term based on summarized distribution
  - Penalize on inverted trigger that is out of range



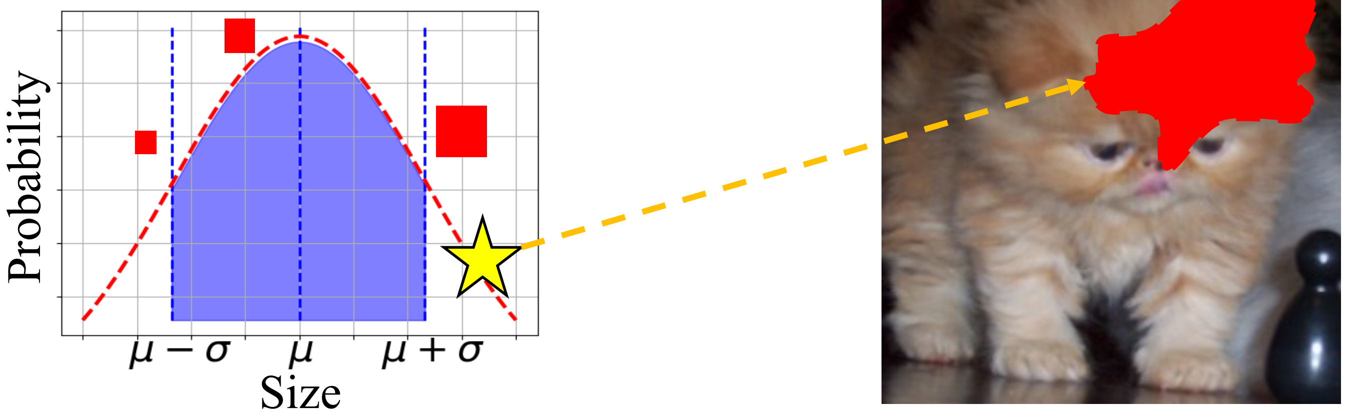
Phase III: Scanner Synthesis

$$Loss = Loss_{ce} + \boxed{Loss_{reg}}$$

- General loss for trigger inversion
  - CE (Cross Entropy) loss ensures target misclassification
  - Reg (Regularization) loss constrains trigger pattern
- Synthesize the regularization term based on summarized distribution
  - Penalize on inverted trigger that is out of range

Trigger Inversion

Distribution

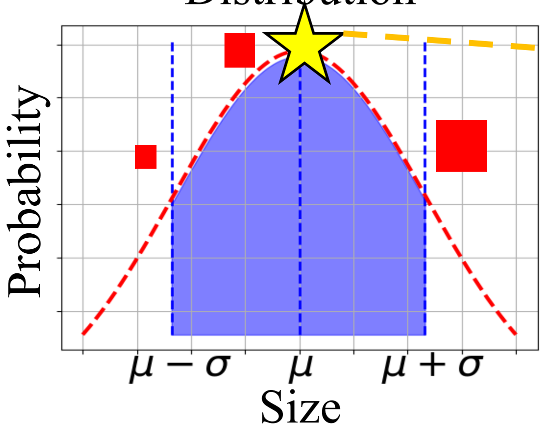


Phase III: Scanner Synthesis

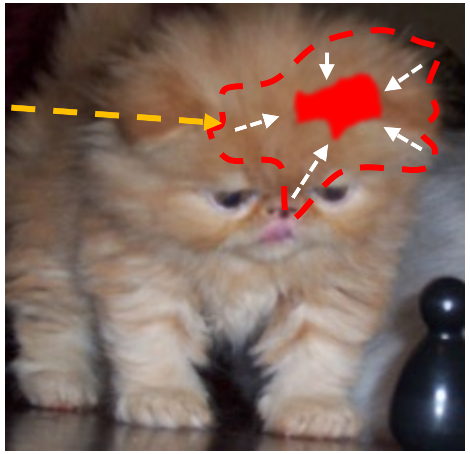
$$Loss = Loss_{ce} + \boxed{Loss_{reg}}$$

- General loss for trigger inversion
  - CE (Cross Entropy) loss ensures target misclassification
  - Reg (Regularization) loss constrains trigger pattern
- Synthesize the regularization term based on summarized distribution
  - Penalize on inverted trigger that is out of range

Distribution



Trigger Inversion



Other application

- Backdoor mitigation
  - Stamp the decomposed trigger on the clean images and perform adversarial training to mitigate the backdoor effect

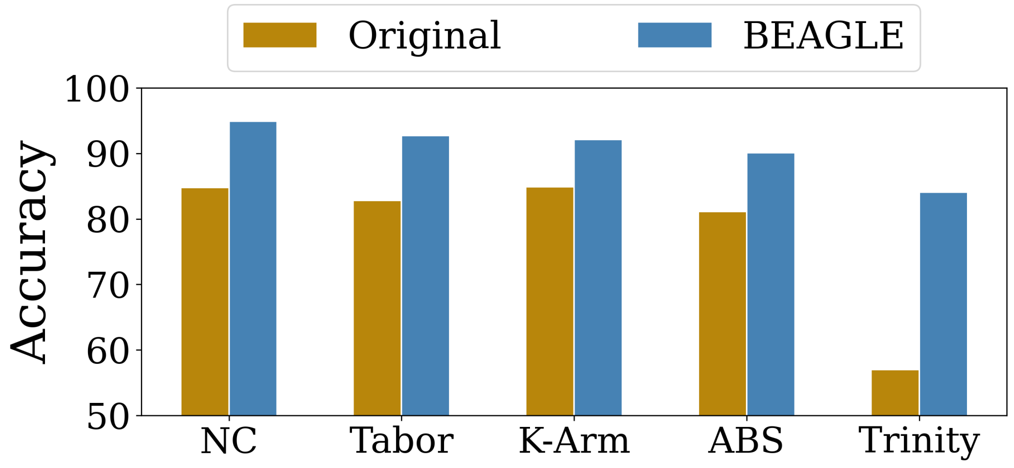
Experiment Setup

- Datasets and models
  - Datasets: TrojAI[1] round 2, 3, CIFAR-10, GTSRB, CelebA, ImageNet
  - Models: ResNet18, ResNet50, VGG11, VGG16, MobileNet, DenseNet…
  - 2112 downloaded models + 420 pre-trained models
- Baselines
  - 10 popular backdoor attacks
  - Improve 5 existing trigger-inversion based backdoor scanners

[1] "Trojai leaderboard," https://pages.nist.gov/trojai/.

Evaluation on Enhanced Scanner

- Metrics: Accuracy, FPR, FNR
- Downloaded TrojAI models
  - Improves NC[1], Tabor[2] and K-Arm[3] for 10% accuracy on polygon backdoored models
  - Improve ABS[4] and Trinity[5] for 9%-27% accuracy on Instagram filter backdoored models
- Pre-trained models
  - Improve ABS[4] for 17% to 40% accuracy on 10 popular backdoored models



[1] Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." S&P 2019.
[2] Guo, Wenbo, et al. "Towards Inspecting and Eliminating Trojan Backdoors in Deep Neural Networks." ICDM 2020.
[3] Shen, Guangyu, et al. "Backdoor scanning for deep neural networks through k-arm optimization." ICML 2021.
[4] Liu, Yingqi, et al. "Abs: Scanning neural networks for back-doors by artificial brain stimulation." CCS 2019.
[5] Karan Sikka, et al. "Detecting Trojaned DNNs Using Counterfactual Attributions." arXiv preprint arXiv:2012.02275 (2020).

Evaluation on Attack Decomposition

- Attack decomposition of Reflection[1] backdoor

≈

Trojaned Image | Source Clean | Decomposed Clean

≈

Clean image ⊕ Ground-truth Trigger | Clean image ⊕ Decomposed Trigger

[1] Liu, Yunfei, et al. "Reflection backdoor: A natural backdoor attack on deep neural networks" ECCV 2020.

Evaluation on Attack Decomposition

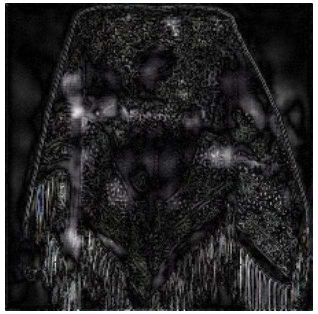
➢ Attack decomposition of Invisible[1] backdoor


Trojaned Image


Source Clean

≈


Decomposed Clean


Ground-truth Trigger

≈


Decomposed Trigger

[1] Li, Yuezun, et al. “Invisible backdoor attack with sample-specific triggers.” ICCV 2021.

Related Work

[1] Gu, Tianyu, et al. “Badnets: Evaluating backdooring attacks on deep neural networks.” IEEE Access 7 (2019).

[2] Salem, Ahmed, et al. “Dynamic backdoor attacks against machine learning models.” EuroS&P 2022.

[3] Chen, Xinyun, et al. “Targeted backdoor attacks on deep learning systems using data poisoning.” arXiv:1712.05526 (2017).

[4] Nguyen, Tuan A, et al. “WaNet-Imperceptible Warping-based Backdoor Attack.” ICLR 2021.

[5] Liu, Yunfei, et al. “Reflection backdoor: A natural backdoor attack on deep neural networks” ECCV 2020.

[6] Li, Yuezun, et al. “Invisible backdoor attack with sample-specific triggers.” ICCV 2021.

[7] Wang, Bolun, et al. “Neural cleanse: Identifying and mitigating backdoor attacks in neural networks.” IEEE S&P 2019.

[8] Guo, Wenbo, et al. “Towards Inspecting and Eliminating Trojan Backdoors in Deep Neural Networks.” ICDM 2020.

[9] Shen, Guangyu, et al. “Backdoor scanning for deep neural networks through k-arm optimization.” ICML 2021.

[10] Karan Sikka, et al. “Detecting Trojaned DNNs Using Counterfactual Attributions.” arXiv:2012.02275 (2020).

[11] Liu, Yingqi, et al. “Abs: Scanning neural networks for back-doors by artificial brain stimulation.” CCS 2019.

[12] TrojAI Leaderboard, https://pages.nist.gov/trojai/

……

Conclusion

- Propose a novel Backdoor Forensics Technique (BEAGLE) can extract the trigger features from the trojaned images and guide trigger inversion.
- BEAGLE can improve scanning accuracy for 10%-16% on average on downloaded TrojAI models and 17%-40% on 10 popular backdoors
- BEAGLE can decompose a trojaned image into its clean version and the trigger with high reconstruction quality.

Thank you!

Q&A