Trigger Hunting with a Topological Prior for Trojan Detection

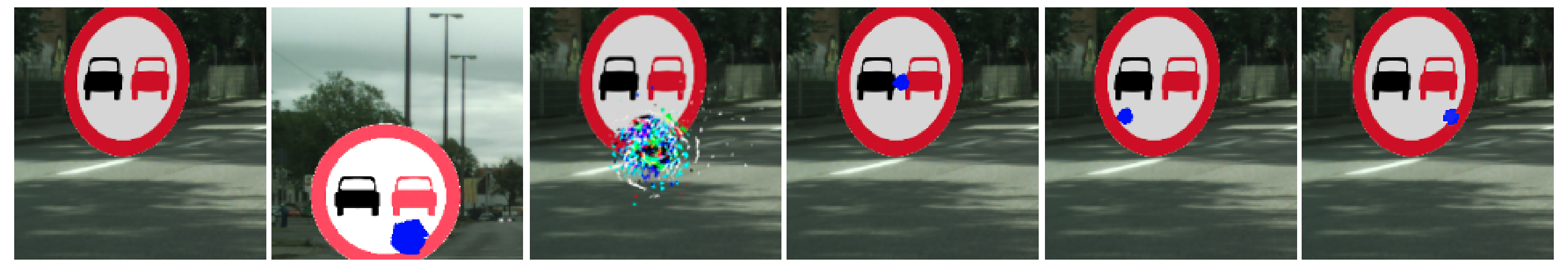
Xiaoling Hu[1], Xiao Lin[2], Michael Cogswell[2], Yi Yao[2], Susmit Jha[2] and Chao Chen[1]

1, Stony Brook University, USA 2, SRI International, USA

1 Introduction

Problem: Find a classifier to distinguish clean models and Trojaned models

- 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



Contribution: a reverse-engineering approach.

- Diversity loss: Trigger candidates different from each other
- Topological constraint: the trigger is a single component
- Improve both classification accuracy and recovered trigger quality

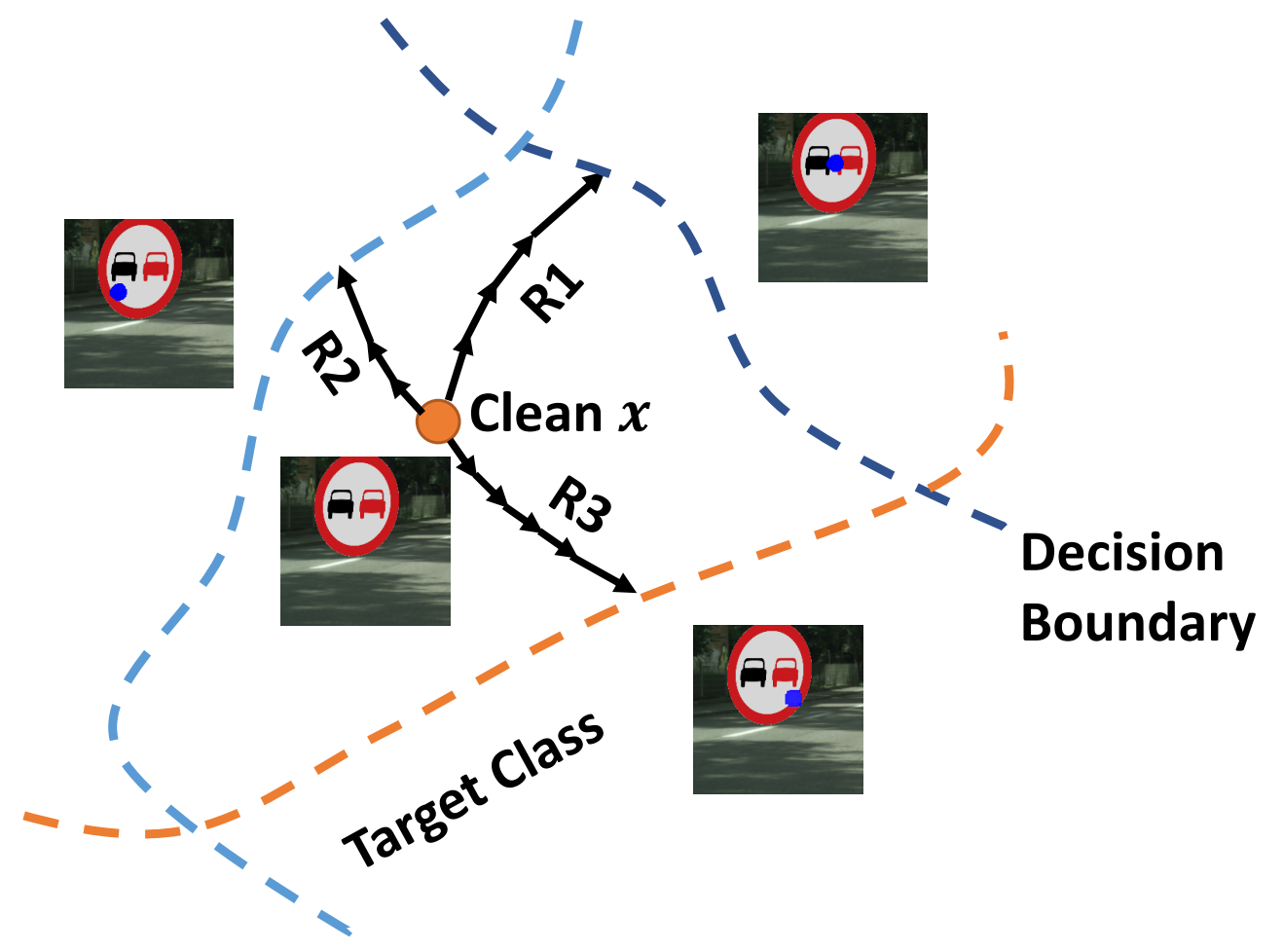
2 Method

Challenge: Limited-data setting: only a few clean samples per class Clean and Trojaned models perform the same on them, and if Trojaned, trigger (location, shape, color) is **unknown**

Key: topological loss, diversity loss in reverse engineering to reduce the huge search space of reverse engineering approach

Intuition of diversity loss

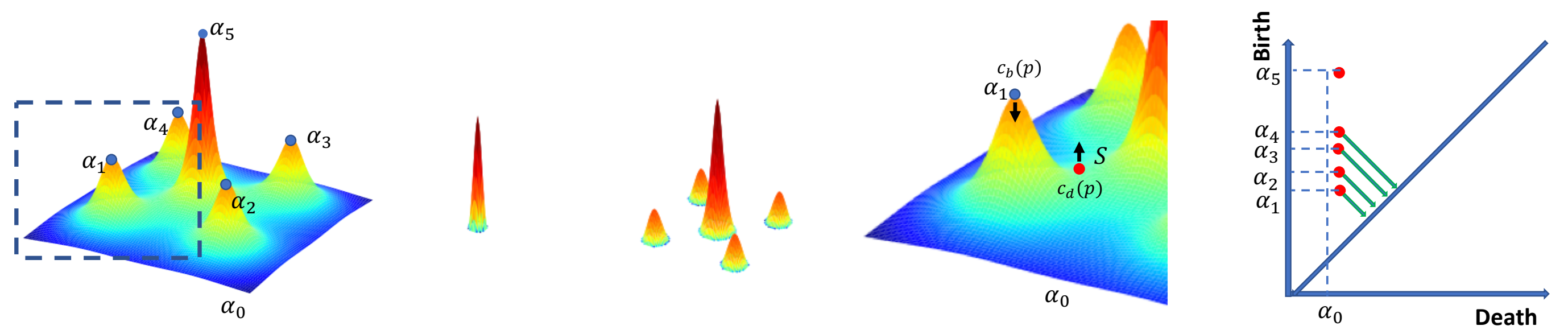
Instead of running gradient decent once, we run it for multiple rounds, each time producing one trigger candidate. Furthermore, we propose a *trigger diversity loss* to ensure the trigger candidates to be sufficiently different from each other. Generating multiple diverse trigger candidates can increase the chance of finding the true trigger. It also mitigates the risk of unknown target labels.



Diversity loss:

$$L_{div}(\mathbf{m},\boldsymbol{\theta}) = -\sum_{j=1}^{i-1} ||\mathbf{m}\odot\boldsymbol{\theta} - \mathbf{m}_j\odot\boldsymbol{\theta}_j||_2.$$

Illustration of persistence diagram



Topological loss and the gradient

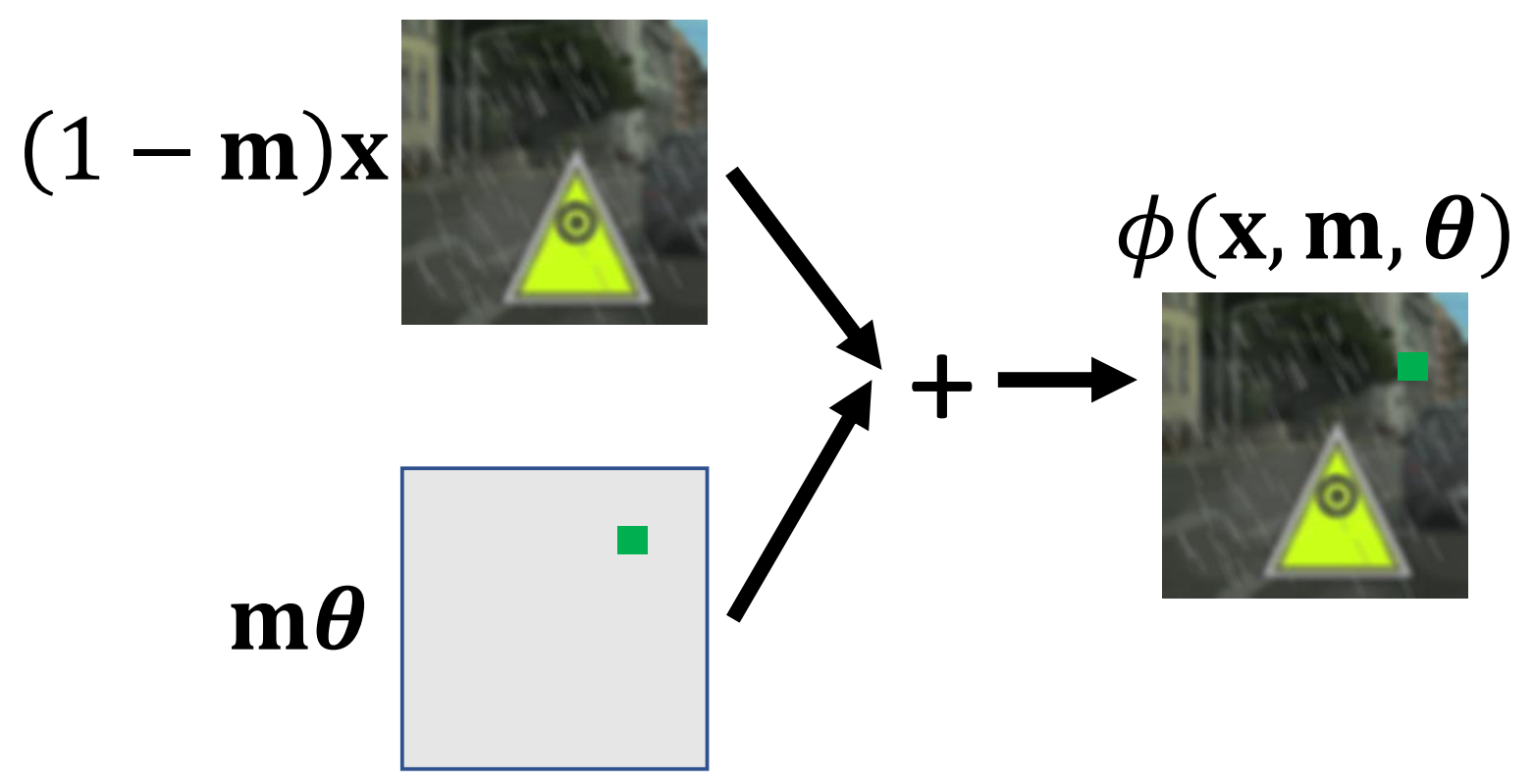
We use the tool of persistent homology to force the recovered trigger to be a single connected component, which can reduce the search space

$$L_{topo}(\mathbf{m}) = \sum_{p\in \mathrm{Dgm}(m)\setminus\{p^*\}} [\mathrm{birth}(p) - \mathrm{death}(p)]^2$$

$$L_{topo}(\mathbf{m}) = \sum_{p\in \mathrm{Dgm}(m)\setminus\{p^*\}} [\mathbf{m}(c_b(p)) - \mathbf{m}(c_d(p))]^2$$

Reverse engineering pipeline

We introduce parameters $\boldsymbol{\theta}$ and $\mathbf{m}$ to convert $\mathbf{x}$ into an altered sample $\phi(\mathbf{x},\mathbf{m},\boldsymbol{\theta}) = (\mathbf{1}-\mathbf{m})\odot\mathbf{x}+\mathbf{m}\odot\boldsymbol{\theta}$ where the binary mask $\mathbf{m} \in \{0,1\}^{M\times N}$ and the pattern $\boldsymbol{\theta} \in \mathbb{R}^{M\times N}$ determine the trigger. $\mathbf{1}$ denotes an all-one matrix. The symbol "$\odot$" denotes Hadamard product.

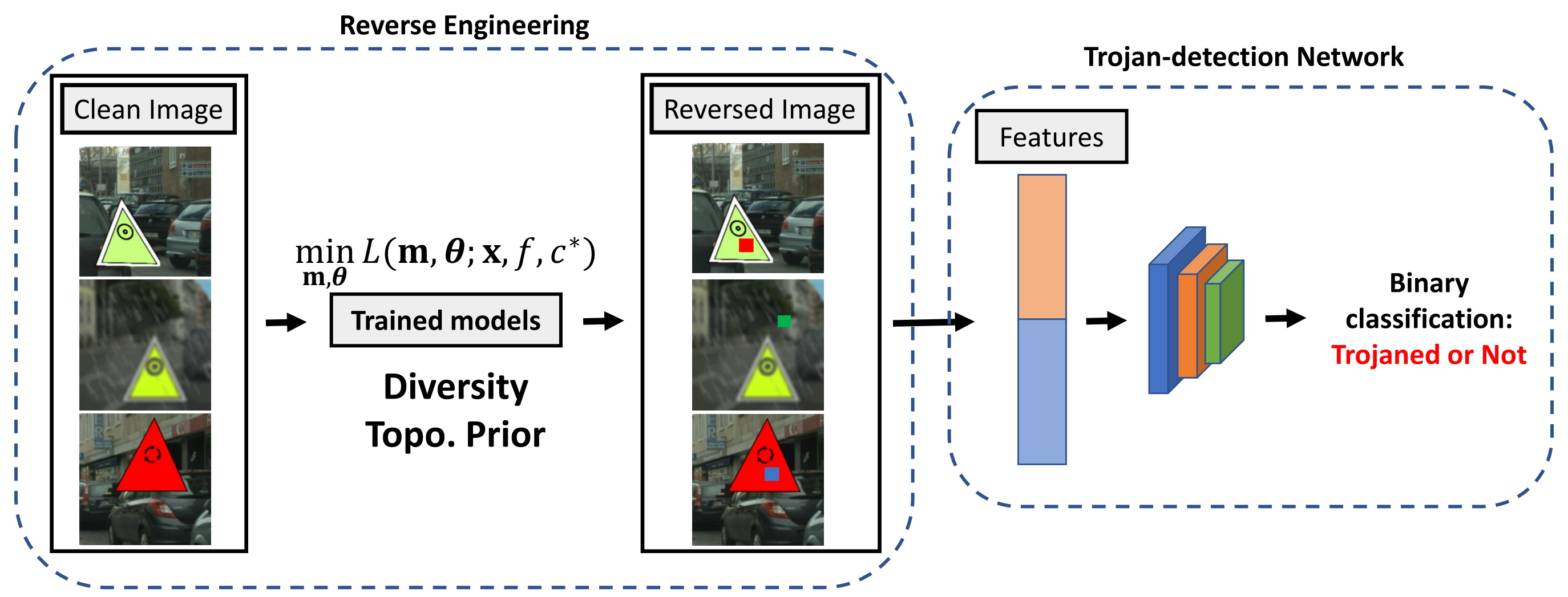

$(1-\mathbf{m})\mathbf{x}$
$\mathbf{m}\boldsymbol{\theta}$
+
$\phi(\mathbf{x},\mathbf{m},\boldsymbol{\theta})$

Label-flipping loss L_{flip} penalizes the prediction of the model regarding the ground truth label, formally:

Our final loss function

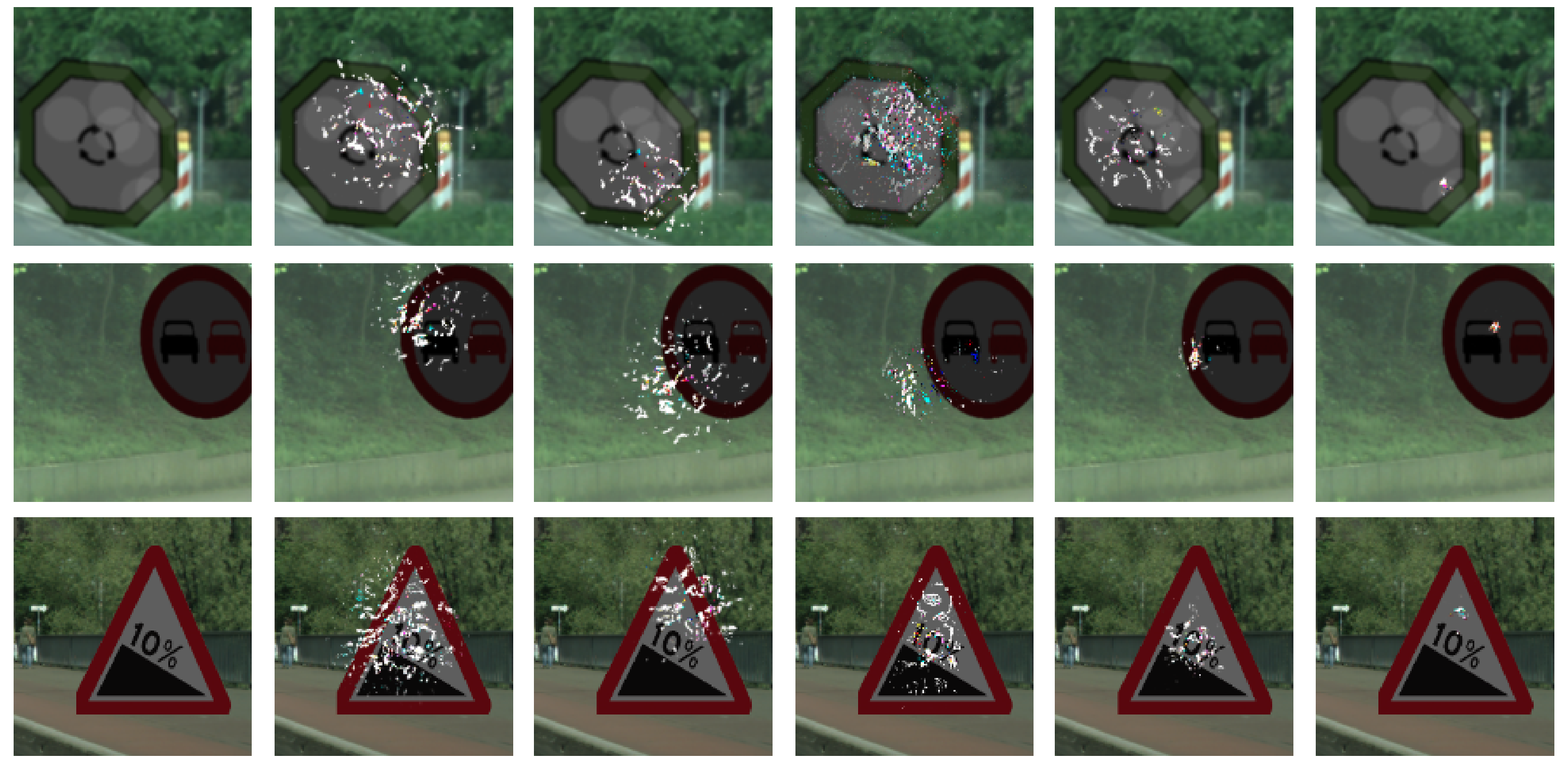
$$L_{flip}(\ldots) + \lambda_1 L_{div}(\ldots) + \lambda_2 L_{topo}(\ldots) + R(\mathbf{m})$$

Workflow of our algorithm



3 Results

Qualitative results



- This shows the qualitative results compared with several popular methods, such as Neural Cleanse, ABS, TABOR and w/o the topological loss.
- From the results, we could find that our proposed method recovers more compact triggers.

Quantitative Results for trojaAI benchmarks

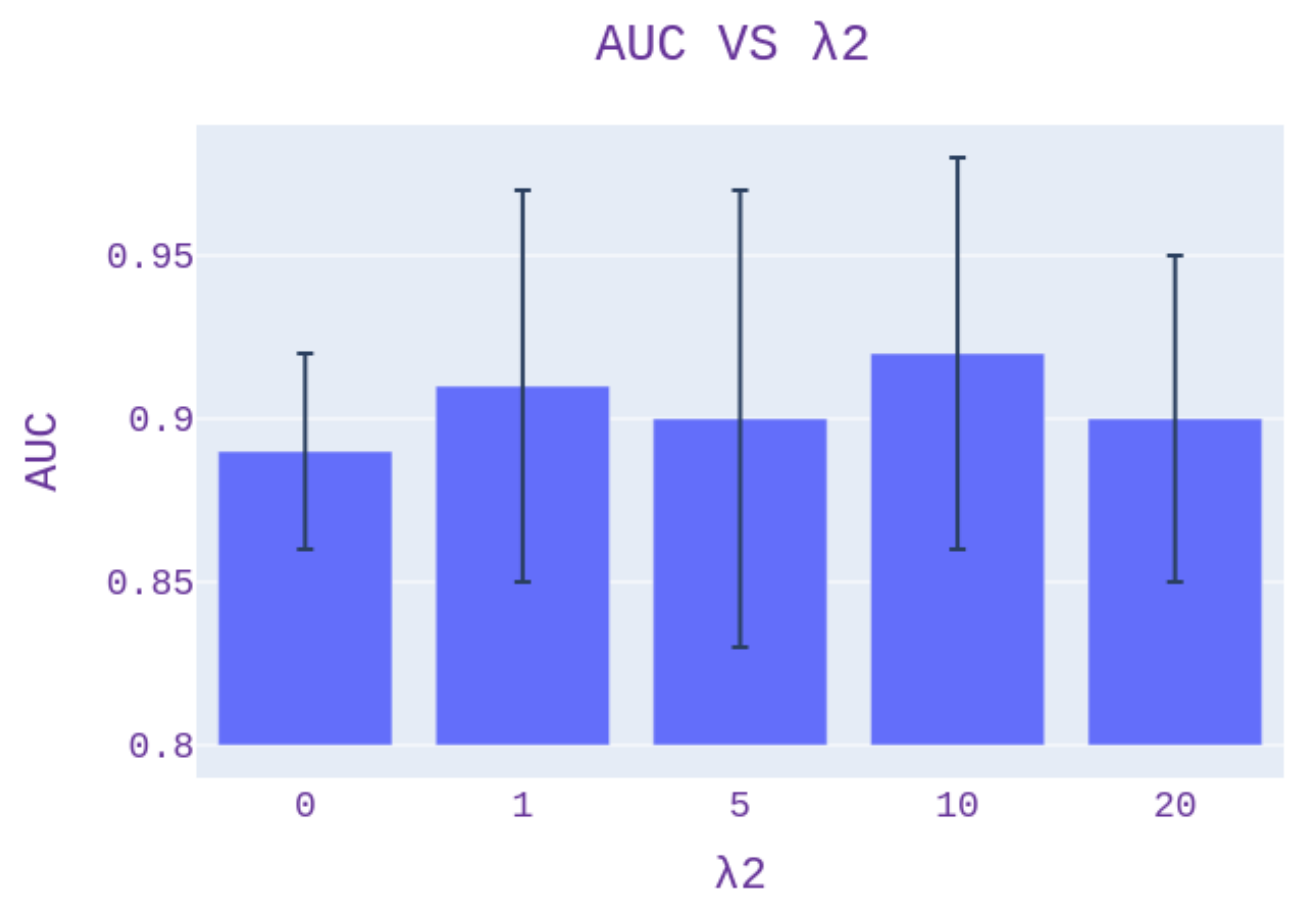
- These datasets are provided by US IARPA/NIST3, who recently organized a Trojan AI competition. Polygon triggers are generated randomly with variations in shape, size, and color.
- We evaluate our approach on the whole set by doing an 8-fold cross validation. For each fold, we use 80% of the models for training, 10% for validation, and the rest 10% for testing.
- Our method achieves superior performance on both the AUC and ACC.

Method	Metric	TrojAI-Round1	TrojAI-Round2	TrojAI-Round3	TrojAI-Round4
NC	AUC	0.50 ± 0.03	0.63 ± 0.04	0.61 ± 0.06	0.58 ± 0.05
ABS	AUC	0.68 ± 0.05	0.61 ± 0.06	0.57 ± 0.04	0.53 ± 0.06
TABOR	AUC	0.71 ± 0.04	0.66 ± 0.07	0.50 ± 0.07	0.52 ± 0.04
ULP	AUC	0.55 ± 0.06	0.48 ± 0.02	0.53 ± 0.06	0.54 ± 0.02
DLTND	AUC	0.61 ± 0.07	0.58 ± 0.04	0.62 ± 0.07	0.56 ± 0.05
Ours	AUC	$\mathbf{0.90 \pm 0.02}$	$\mathbf{0.87 \pm 0.05}$	$\mathbf{0.89 \pm 0.04}$	$\mathbf{0.92 \pm 0.06}$
NC	ACC	0.53 ± 0.04	0.49 ± 0.02	0.59 ± 0.07	0.60 ± 0.04
ABS	ACC	0.70 ± 0.04	0.59 ± 0.05	0.56 ± 0.03	0.51 ± 0.05
TABOR	ACC	0.70 ± 0.03	0.68 ± 0.08	0.51 ± 0.05	0.55 ± 0.06
ULP	ACC	0.58 ± 0.07	0.51 ± 0.03	0.56 ± 0.04	0.57 ± 0.04
DLTND	ACC	0.59 ± 0.04	0.61 ± 0.05	0.65 ± 0.04	0.59 ± 0.06
Ours	ACC	$\mathbf{0.91 \pm 0.03}$	$\mathbf{0.89 \pm 0.04}$	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.91 \pm 0.04}$

Quantitative Results on Trojaned-MNIST/CIFAR10

Method	Metric	Trojaned-MNIST	Trojaned-CIFAR10
NC	AUC	0.57 ± 0.07	0.75 ± 0.07
ABS	AUC	0.63 ± 0.04	0.67 ± 0.06
TABOR	AUC	0.65 ± 0.07	0.71 ± 0.05
ULP	AUC	0.59 ± 0.03	0.55 ± 0.03
DLTND	AUC	0.62 ± 0.05	0.52 ± 0.08
Ours	AUC	$\mathbf{0.88 \pm 0.04}$	$\mathbf{0.91 \pm 0.05}$
NC	ACC	0.60 ± 0.04	0.73 ± 0.06
ABS	ACC	0.65 ± 0.02	0.69 ± 0.04
TABOR	ACC	0.62 ± 0.04	0.69 ± 0.08
ULP	ACC	0.57 ± 0.02	0.59 ± 0.06
DLTND	ACC	0.64 ± 0.07	0.55 ± 0.07
Ours	ACC	$\mathbf{0.89 \pm 0.02}$	$\mathbf{0.92 \pm 0.04}$

Ablation study for loss weights. For the loss weights λ_1 and λ_2, we empirically choose the weights which make reverse engineering converge the fastest. This is a reasonable choice as in practice, time is one major concern for reverse engineering pipelines.



Ablation study: Number of training samples and loss terms

# of samples	Ours	w/o topo	w/o diversity
25	$\mathbf{0.77 \pm 0.04}$	0.73 ± 0.03	0.68 ± 0.04
50	$\mathbf{0.81 \pm 0.03}$	0.76 ± 0.05	0.73 ± 0.02
100	$\mathbf{0.84 \pm 0.05}$	0.78 ± 0.06	0.76 ± 0.03
200	$\mathbf{0.86 \pm 0.04}$	0.82 ± 0.04	0.79 ± 0.05
400	$\mathbf{0.90 \pm 0.05}$	0.85 ± 0.03	0.82 ± 0.04
800	$\mathbf{0.92 \pm 0.06}$	0.89 ± 0.04	0.85 ± 0.02

Method	TrojAI-Round4
w/o topological loss	0.89 ± 0.04
w/o diversity loss ($N_T = 1$)	0.85 ± 0.02
$N_T = 2$	0.89 ± 0.05
with all loss terms ($N_T = 3$)	$\mathbf{0.92 \pm 0.06}$

Acknowledgement

The authors thank anonymous reviewers for their constructive feedback. This effort was partially supported by the Intelligence Advanced Research Projects Agency (IARPA) under the contract W911NF20C0038. The content of this paper does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.