David J. Miller

Zhen Xiang

George Kesidis

Adversarial Learning and Secure AI

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

Chapter 10

Test-Time Detection of Backdoor Triggers

Outline

1. Test-Time (In-Flight) Backdoor Defense Scenario
2. Defenses
3. In-Flight RED (IF-RED)
4. Experiments

Test-time Backdoor Defense Scenario

- Do not assume access to the training dataset
- Assume access to the model and to a small clean dataset that could be used to determine whether the model was backdoor poisoned
- Some leverage an unsupervised PT-RED’s estimate of the backdoor pattern
- Do not assume availability of ground-truth examples of backdoor-poisoned samples (backdoor triggers) from the attacker

STRIP and SentiNet (cont)

- STRIP (implemented in IBM ART) & SentiNet defenses
  - Goal is to detect test-time samples that trigger the backdoor
  - Assume that the AI has already been detected as backdoor poisoned (!)
  - Assume defender is given a pool of samples known to trigger the backdoor, and a pool of (clean) samples that do not trigger the backdoor (!)

In Flight RED (IF-RED)

Employs effective REDs to estimate backdoor patterns, e.g., see Chapters 6 and 7.

The true backdoor pattern and its estimate will likely activate the same set of neurons (when they are embedded in images from the same source class).

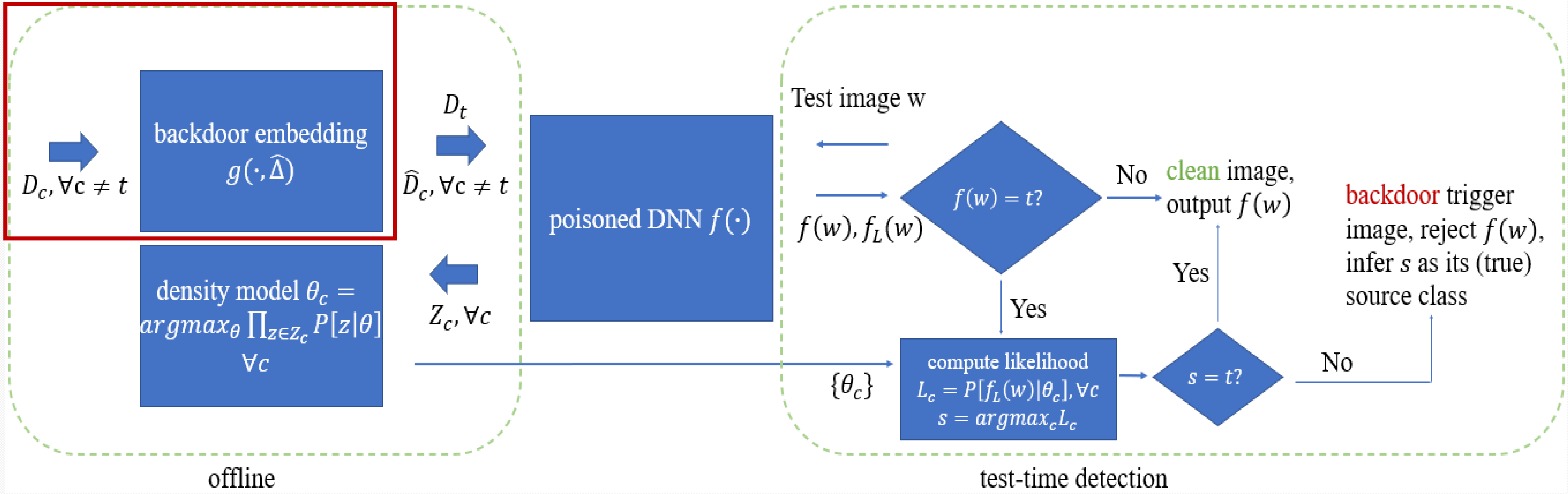
These neurons are likely different from those activated by typical target class images.

If an image classified to the target class t is a backdoor trigger image, then its deep layer activations are expected to be

- similar to the activations for most images from the same source class embedded with the estimated pattern;
- different from the activations for typical images from the target class.

IF-RED: Step 1

Step1. Embed estimated backdoor pattern in clean images of non-target classes,
$\widehat{D}_c = \{g(x, \widehat{\Delta}) | x \in D_c\}, \forall c \neq t.$

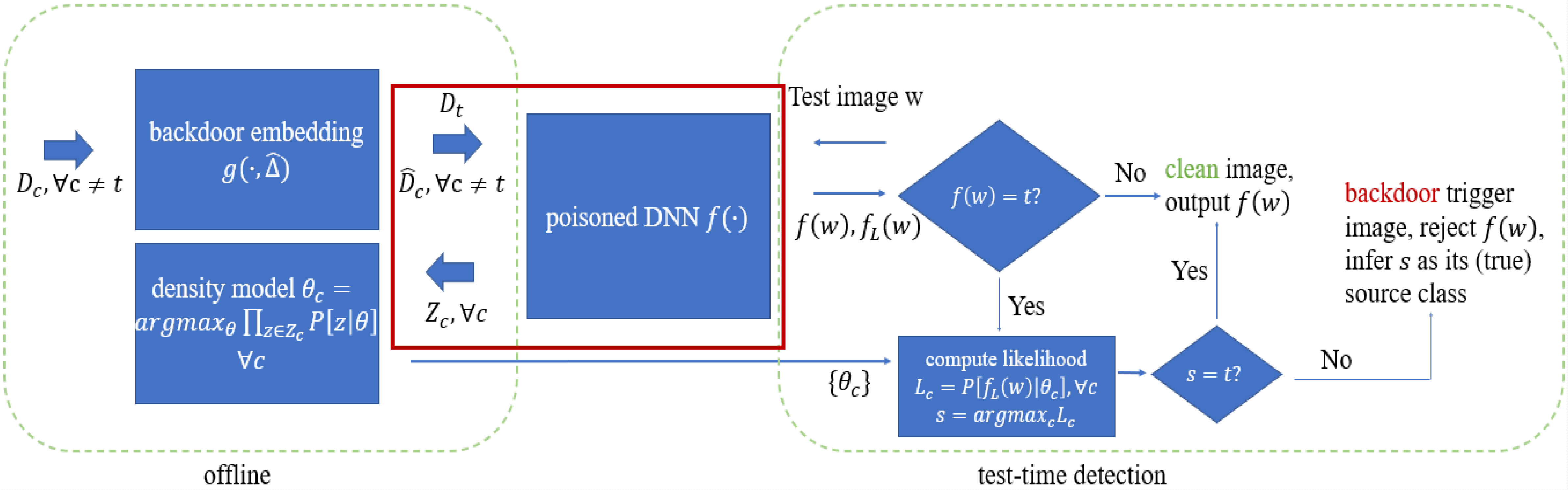

$D_c, \forall c \neq t$
backdoor embedding
$g(\cdot, \widehat{\Delta})$
D_t
$\widehat{D}_c, \forall c \neq t$
density model $\theta_c = argmax_\theta \prod_{z \in Z_c} P[z|\theta]$
$\forall c$
$Z_c, \forall c$
poisoned DNN $f(\cdot)$
Test image w
$f(w), f_L(w)$
$f(w) = t?$
No
clean image, output $f(w)$
Yes
Yes
$\{\theta_c\}$
compute likelihood
$L_c = P[f_L(w)|\theta_c], \forall c$
$s = argmax_c L_c$
$s = t?$
No
backdoor trigger image, reject $f(w)$, infer s as its (true) source class
offline
test-time detection

IF-RED: Step 2

Step 2. Feed samples in $\cup_{c \neq t} \widehat{D}_c$ and D_t into DNN and get their internal layer (layer L) features

$Z_c = \{f_L(\hat{x}) | \hat{x} \in \widehat{D}_c\}, \forall c \neq t,$

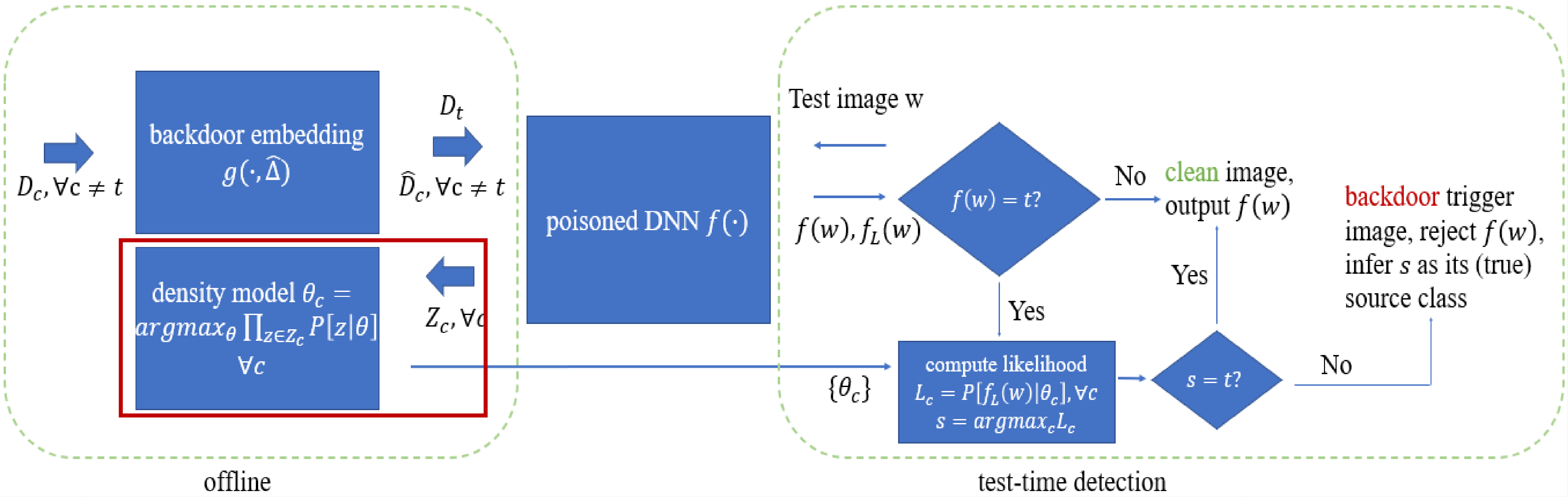
$Z_t = \{f_L(x) | x \in D_t\}.$



$D_c, \forall c \neq t$
backdoor embedding $g(\cdot, \widehat{\Delta})$
density model $\theta_c = argmax_\theta \prod_{z \in Z_c} P[z|\theta]$ $\forall c$
D_t
$\widehat{D}_c, \forall c \neq t$
$Z_c, \forall c$
poisoned DNN $f(\cdot)$
Test image w
$f(w), f_L(w)$
$f(w) = t?$
No
clean image, output $f(w)$
Yes
Yes
backdoor trigger image, reject $f(w)$, infer s as its (true) source class
$\{\theta_c\}$
compute likelihood $L_c = P[f_L(w)|\theta_c], \forall c$ $s = argmax_c L_c$
$s = t?$
No
offline
test-time detection

IF-RED: Step 3

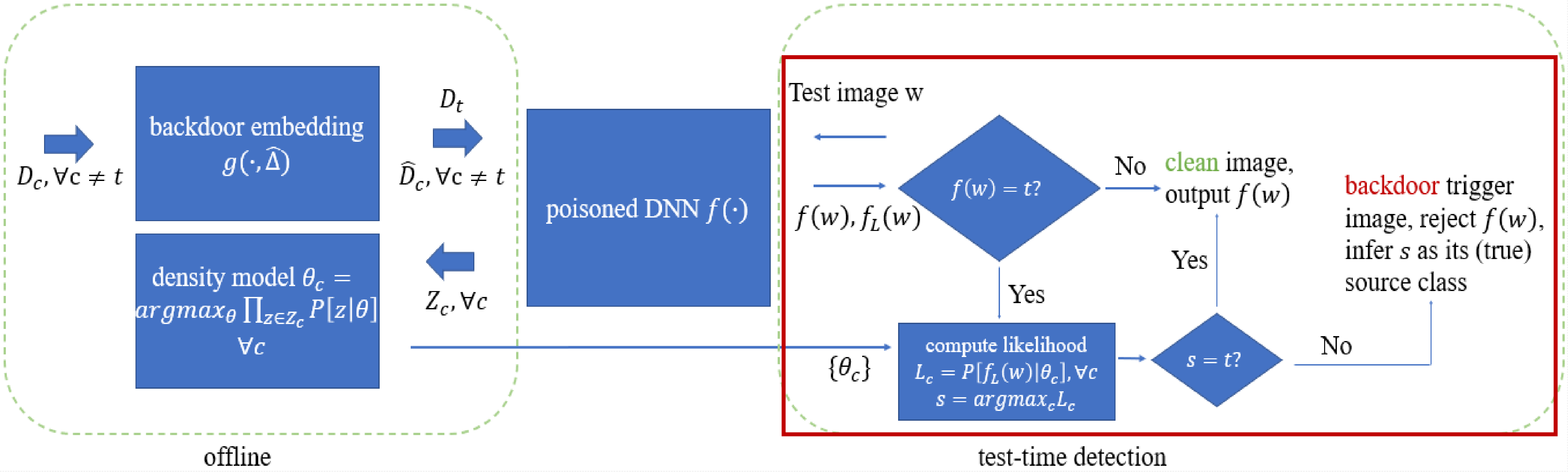
Step 3. Learn a density model for each class c on its normalized internal layer features:
$\theta_c = \text{argmax}_\theta \prod_{z \in Z_c} P[z|\theta]$.



$D_c, \forall c \neq t$
backdoor embedding $g(\cdot, \hat{\Delta})$
D_t
$\hat{D}_c, \forall c \neq t$
density model $\theta_c = argmax_\theta \prod_{z \in Z_c} P[z|\theta]$ $\forall c$
$Z_c, \forall c$
poisoned DNN $f(\cdot)$
Test image w
$f(w), f_L(w)$
$f(w) = t?$
No
clean image, output $f(w)$
Yes
backdoor trigger image, reject $f(w)$, infer s as its (true) source class
Yes
$\{\theta_c\}$
compute likelihood $L_c = P[f_L(w)|\theta_c], \forall c$ $s = argmax_c L_c$
$s = t?$
No
offline
test-time detection

Method: Step 4

Step 4. For a test image w with $f(w) = t$, we measure its likelihood $L_c = P[f_L(w)|\theta_c], \forall c$.
If $s = \text{argmax}_c L_c \neq t$, w is deemed to contain the backdoor pattern, and infer s as its source class; otherwise, w is deemed clean and we accept the DNN's class prediction.


$D_c, \forall c \neq t$
backdoor embedding $g(\cdot, \hat{\Delta})$
D_t
$\hat{D}_c, \forall c \neq t$
density model $\theta_c = argmax_\theta \prod_{z \in Z_c} P[z|\theta]$ $\forall c$
$Z_c, \forall c$
poisoned DNN $f(\cdot)$
Test image w
$f(w), f_L(w)$
$f(w) = t?$
No
clean image, output $f(w)$
Yes
Yes
$\{\theta_c\}$
compute likelihood $L_c = P[f_L(w)|\theta_c], \forall c$ $s = argmax_c L_c$
$s = t?$
No
backdoor trigger image, reject $f(w)$, infer s as its (true) source class
offline
test-time detection

Experimental Set-Up on CIFAR-10

Dataset: CIFAR-10, 10 classes, 5k training images per class, 1k test images per class

Target DNN: ResNet-18

Data allocation: split the test set – 100 images per class are preserved for the defender, the remaining are used for testing.

Attack settings: We create the attacks following the threat model.

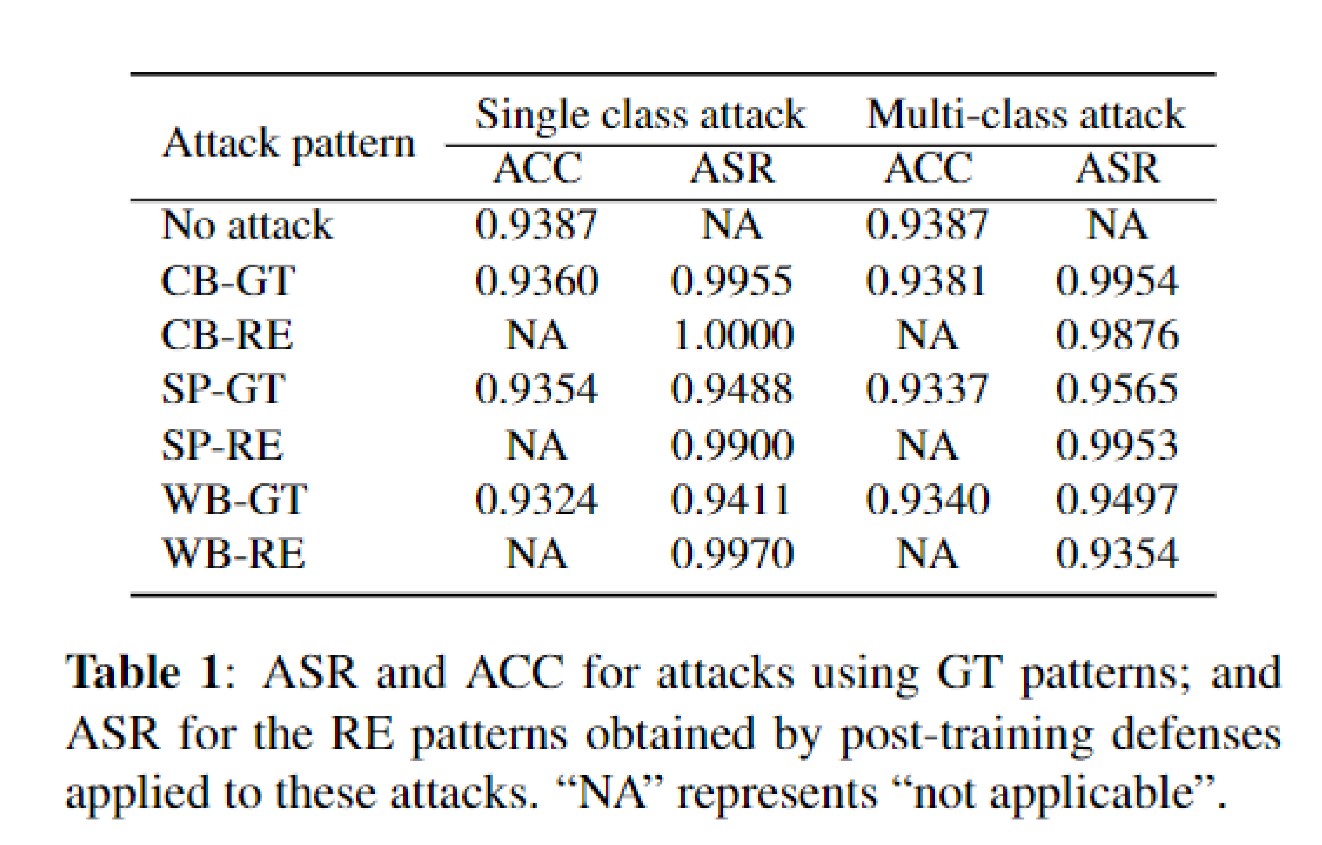
- Target class: class 9
- Three backdoor patterns: 1) an additive perturbation "chess board" (CB); 2) a random single pixel set to 255 (SP); and 3) a 3×3 white box patch (WB).
- 2 attacks for each pattern: 1) single source-class attack (one-to-one): embed the backdoor pattern in 1000 images of class 0; 2) multi source-class attack (all-to-one): embed the backdoor pattern in 100 images of all non-target classes.
- Embed the backdoor pattern into test images from the source class(es) for test-time attack.

Defense settings: Apply P-PT-RED and I-PT-RED to infer the target class and estimate the backdoor pattern. Learn a Gaussian mixture model on the penultimate layer features for each class.

Experimental Results

We measure the effectiveness of backdoor attacks using ground truth (GT) pattern and reverse-engineered (RE) pattern by

- attack success rate (ASR): the fraction of test images embedded with backdoor pattern that are misclassified to the target class.
- clean test accuracy (ACC): DNN’s accuracy on clean test samples



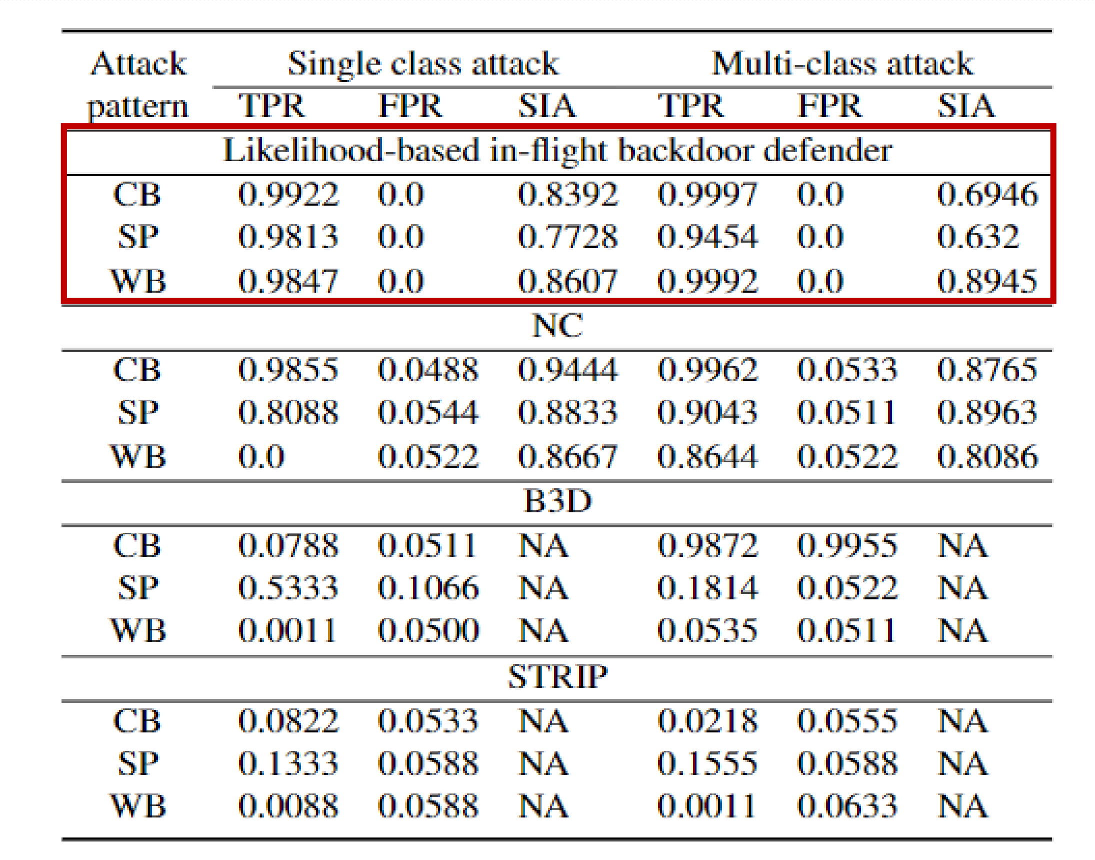
Attack pattern	Single class attack		Multi-class attack	
	ACC	ASR	ACC	ASR
No attack	0.9387	NA	0.9387	NA
CB-GT	0.9360	0.9955	0.9381	0.9954
CB-RE	NA	1.0000	NA	0.9876
SP-GT	0.9354	0.9488	0.9337	0.9565
SP-RE	NA	0.9900	NA	0.9953
WB-GT	0.9324	0.9411	0.9340	0.9497
WB-RE	NA	0.9970	NA	0.9354

Table 1: ASR and ACC for attacks using GT patterns; and ASR for the RE patterns obtained by post-training defenses applied to these attacks. “NA” represents “not applicable”.

Experimental Results

We measure the performance of IF-RED (at top) with NC, B3D, and STRIP by:

- true positive rates (TPR): the fraction of backdoor-trigger images correctly detected;
- false positive rates (FPR): the fraction of clean images falsely detected;
- source class inference accuracy (SIA): the fraction of backdoor-trigger images with correct inference of the source class.



Attack pattern	Single class attack			Multi-class attack		
	TPR	FPR	SIA	TPR	FPR	SIA
Likelihood-based in-flight backdoor defender						
CB	0.9922	0.0	0.8392	0.9997	0.0	0.6946
SP	0.9813	0.0	0.7728	0.9454	0.0	0.632
WB	0.9847	0.0	0.8607	0.9992	0.0	0.8945
NC						
CB	0.9855	0.0488	0.9444	0.9962	0.0533	0.8765
SP	0.8088	0.0544	0.8833	0.9043	0.0511	0.8963
WB	0.0	0.0522	0.8667	0.8644	0.0522	0.8086
B3D						
CB	0.0788	0.0511	NA	0.9872	0.9955	NA
SP	0.5333	0.1066	NA	0.1814	0.0522	NA
WB	0.0011	0.0500	NA	0.0535	0.0511	NA
STRIP						
CB	0.0822	0.0533	NA	0.0218	0.0555	NA
SP	0.1333	0.0588	NA	0.1555	0.0588	NA
WB	0.0088	0.0588	NA	0.0011	0.0633	NA

Table 2: TPR, FPR and SIA for our defense, compared with three other in-flight defenses, NC, B3D, and STRIP, against all the created attacks. "NA" signifies "not applicable".

Experimental Set-Up:PubFig, MNIST/F-MNIST

Datasets:

- MNIST/F-MNIST, 10 classes, 5k training images per class, 1k test images per class
- PubFig, 20 classes, 80 training images per class, 20 test images per class

Target DNN: LeNet5 on MNIST/F-MNIST, VGG-16 on PubFig

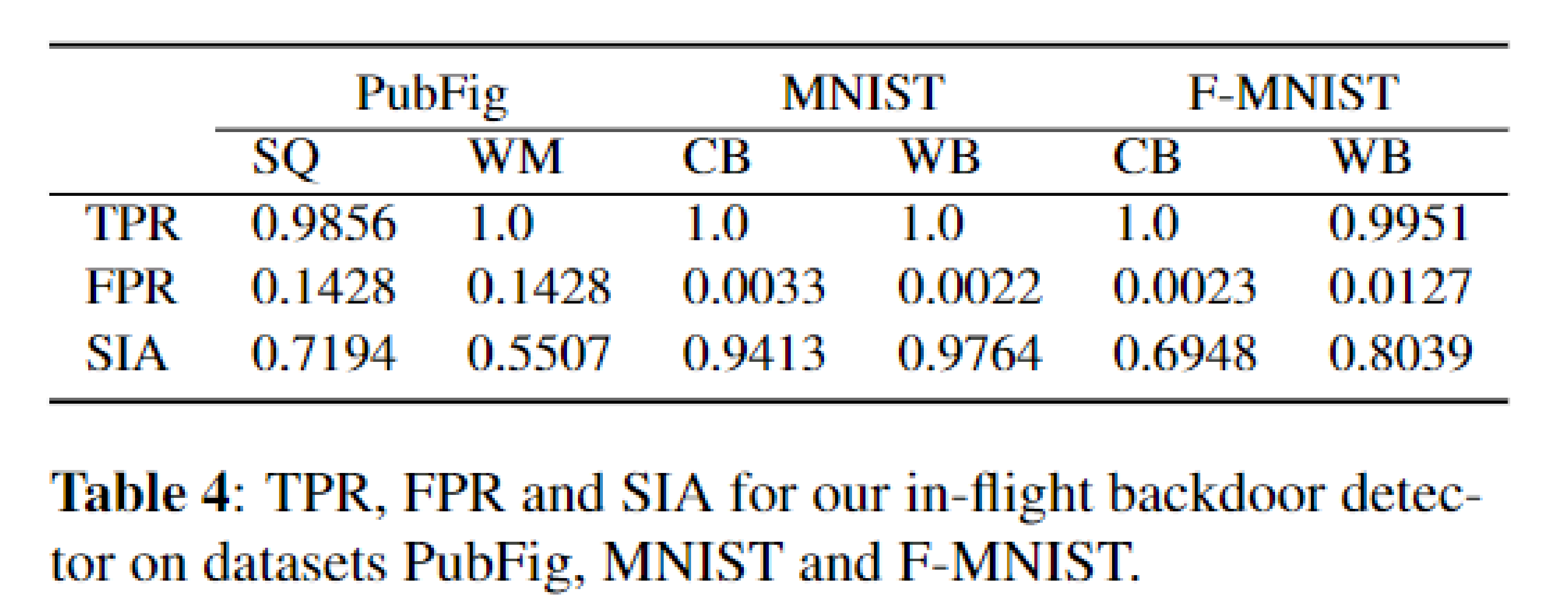
Data allocation: split the test set – 100 images per class for MNIST/F-MINIST, 5 images per class for PubFig, are preserved for the defender, and the remaining are used for testing.

Attack settings:

- Target class: class 9 for MNIST/F-MNIST, class 19 for PubFig
- Backdoor patterns: CB and WB for MNIST/F-MNIST; Trojan square (SQ) and Trojan watermark (WM) for PubFig.
- All-to-one attack: embed the backdoor pattern in 100 images for MNIST/F-MNIST, 2 images for PubFig of all non-target classes.
- Embed the backdoor pattern into test images of the source classes for test-time attack.

Defense settings: same as CIFAR-10

Experimental Results on PubFig, MNIST/F-MNIST



	PubFig		MNIST		F-MNIST	
	SQ	WM	CB	WB	CB	WB
TPR	0.9856	1.0	1.0	1.0	1.0	0.9951
FPR	0.1428	0.1428	0.0033	0.0022	0.0023	0.0127
SIA	0.7194	0.5507	0.9413	0.9764	0.6948	0.8039

Table 4: TPR, FPR and SIA for our in-flight backdoor detector on datasets PubFig, MNIST and F-MNIST.

Other Methods

- Note that IF-RED can also operate with I-PT-RED acting on embedded features.
- One can also employ the “activation clipping” approach of Chapter 9:
    - MMDF considers **both** the original DNN and the one whose activations are bounded (clipped) to mitigate the backdoor.
    - MMDF detects a backdoor trigger if
        - there is disagreement in the inferred class by these two DNNs, or
        - the two DNNs agree but their difference in classification margin is anomalous w.r.t. to a null informed by the clean dataset.

Additional References

- [MMDF] H. Wang, Z. Xiang, D.J. Miller, and G. Kesidis. Improved Activation Clipping for Universal Backdoor Mitigation and Test-Time Detection. https://arxiv.org/abs/2308.04617, 2023.

With Permission, Figures Reproduced From

- X. Li, Z. Xiang, D.J. Miller, G. Kesidis. Test-Time Detection of Backdoor Triggers for Poisoned Deep Neural Networks. In Proc. IEEE ICASSP, Mar. 2022.