# Rethinking Deep Neural Network Ownership Verification: Embedding Passports to Defeat Ambiguity Attacks

Lixin Fan[1], Kam Woh Ng[2], Chee Seng Chan[2]

[1]WeBank AI Lab, Shenzhen, China
[2]Center of Image and Signal Processing, Faculty of Comp. Sci. and Info., Tech.
University of Malaya, Kuala Lumpur, Malaysia

{lixinfan@webank.com;kamwoh@siswa.um.edu.my;cs.chan@um.edu.my}

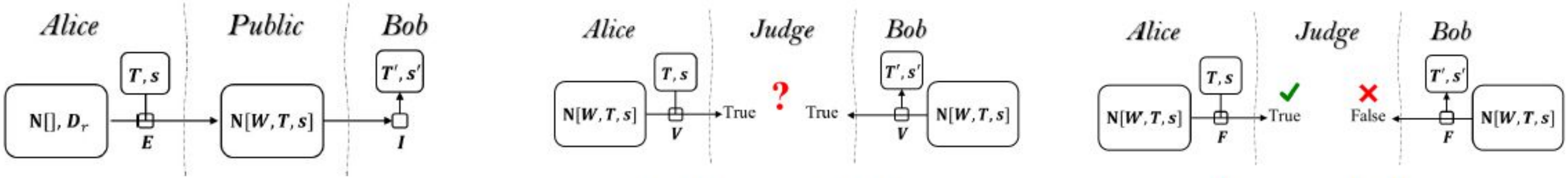
# Table of Content

1. Motivation
2. Application Scenario
3. Ambiguity attack on previous method
4. Proposed Solution: Embedding Passports
5. Effect of proposed solution
6. Passporting Layer
7. Visualization of Passport Layer
8. Embedding Binary Signatures by Sign of Scale Factors
9. Ownership Verification with Passports
10. Experiment Results
11. Network Complexity
12. Conclusion

# Motivation

- More and more business models and services using Deep Neural Network (DNN) such as Machine Learning as a Service (MLaaS)
- We need a **protection on DNN from being illegally copied, distributed and abused**
- We investigate a number of watermark-based DNN ownership verification methods [1, 2, 3] in the face of **ambiguity attacks**, which aim to cast **doubts on ownership verification by forging counterfeit watermarks** and shown that ambiguity attacks pose serious challenges to existing DNN watermarking methods.
- Ambiguity attack: *Hacker embeds a different watermark into the converged model, resulting in both owner and hacker to be able to claim the model's ownership*

# Application Scenario

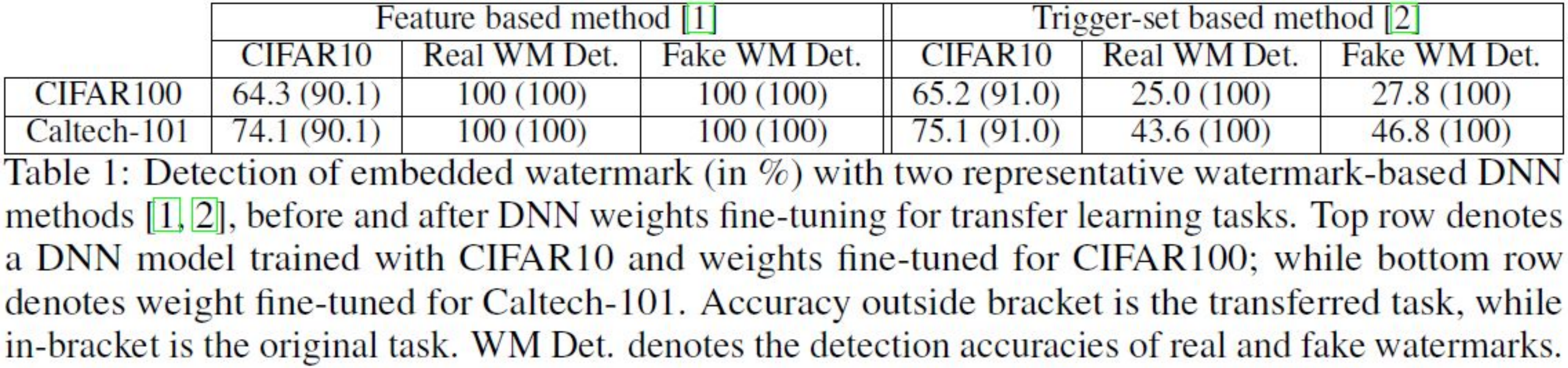


(a) Application Scenario. (b) Present Solution. (c) Proposed Solution.

Figure 1: DNN model ownership verification in the face of ambiguity attacks. **(a)**: Owner *Alice* uses an embedding process $E$ to train a DNN model with watermarks ($\mathbf{T}, \mathrm{s}$) and releases the model publicly available; Attacker *Bob* forges counterfeit watermarks ($\mathbf{T}', \mathrm{s}'$) with an invert process $I$; **(b)**: The ownership is in doubt since both the original and forged watermarks are detected by the verification process $V$ (Sect. 2.2); **(c)**: The ambiguity is resolved when our proposed passports are embedded and the network performances are evaluated in favor of the original passport by the fidelity evaluation process $F$ (See Definition 1 and Sect. 3.3).

*Note: T = trigger set data, s = signature*

# Ambiguity attack on previous method



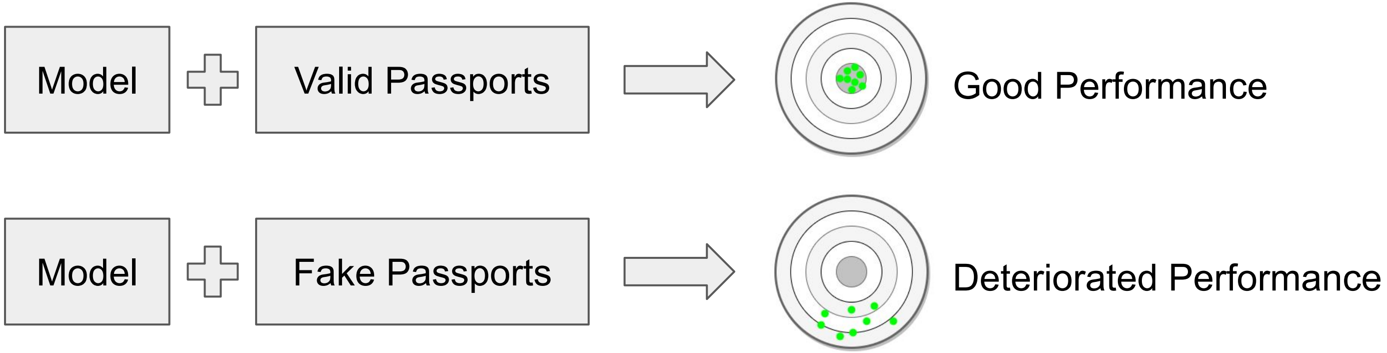
| | Feature based method [1] | | | Trigger-set based method [2] | | |
|---|---|---|---|---|---|---|
| | CIFAR10 | Real WM Det. | Fake WM Det. | CIFAR10 | Real WM Det. | Fake WM Det. |
| CIFAR100 | 64.3 (90.1) | 100 (100) | 100 (100) | 65.2 (91.0) | 25.0 (100) | 27.8 (100) |
| Caltech-101 | 74.1 (90.1) | 100 (100) | 100 (100) | 75.1 (91.0) | 43.6 (100) | 46.8 (100) |

Table 1: Detection of embedded watermark (in %) with two representative watermark-based DNN methods [1, 2], before and after DNN weights fine-tuning for transfer learning tasks. Top row denotes a DNN model trained with CIFAR10 and weights fine-tuned for CIFAR100; while bottom row denotes weight fine-tuned for Caltech-101. Accuracy outside bracket is the transferred task, while in-bracket is the original task. WM Det. denotes the detection accuracies of real and fake watermarks.

1. Feature based (whitebox) - to embed watermark into DNN model using weights in the model, verify the ownership by extracting the watermark features.
2. Trigger-set based (blackbox) - to embed trigger set data into DNN model during the training of the model, verify the ownership by querying API calls and comparing the output results with our expected results

# Proposed Solution: Embedding Passports

- We want to design and train DNN models in a way such that, the network performances of the original task will be significantly deteriorated when forged signatures are used.
- We implemented **Passporting Layers**, followed by different ownership protection schemes that exploit the embedded passports to effectively defeat ambiguity attack



Model + Valid Passports → Good Performance

Model + Fake Passports → Deteriorated Performance

# Effect of proposed solution

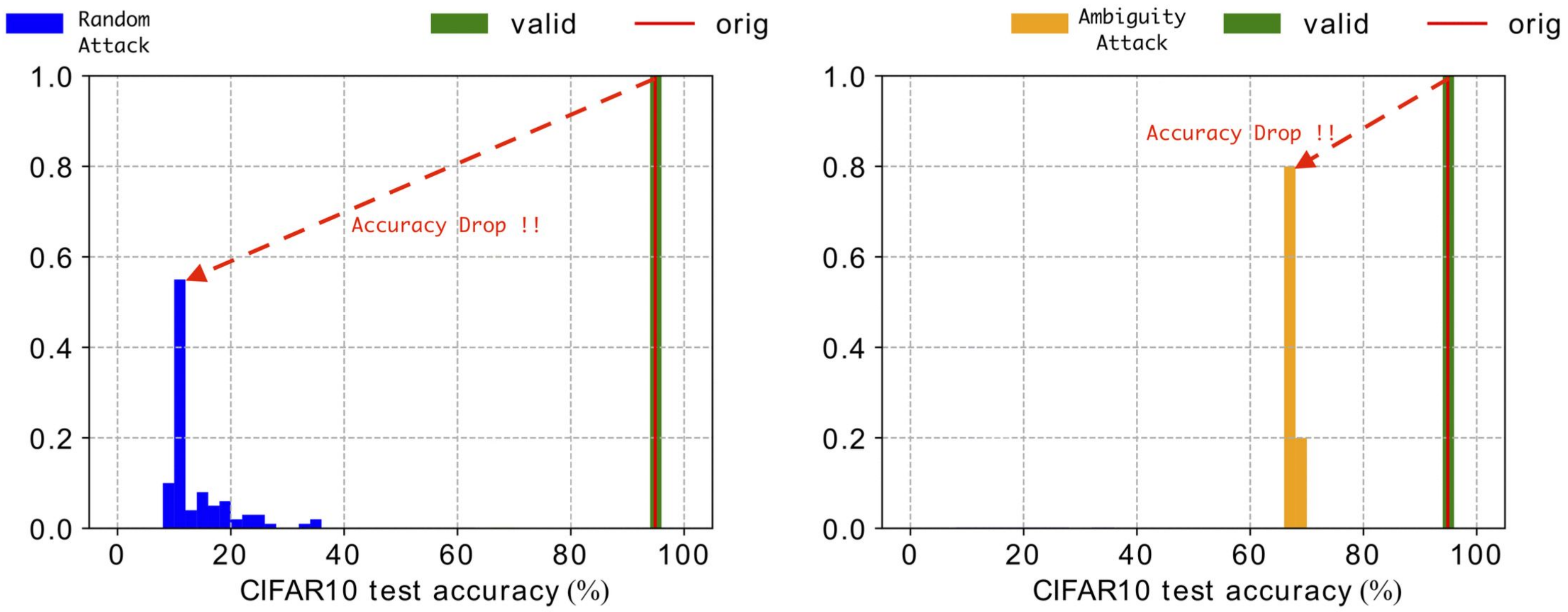
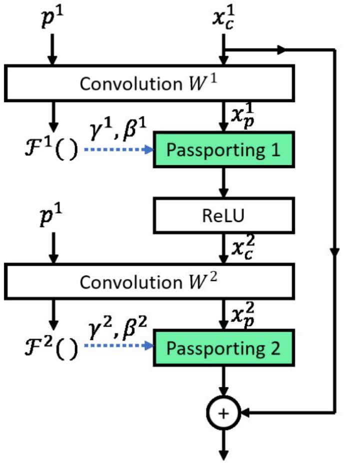


Figure 1: Example of ResNet performance on CIFAR10 when (left) Random Attack and (right) Ambiguity Attack

# Passporting Layer

- In order to **control the network functionalities** by embedded digital signatures i.e. passports, we proposed **to append after a convolution layer a passporting layer**, whose scale factor γ and bias shift term β are dependent on both the convolution kernels Wp and the designated passport P as f

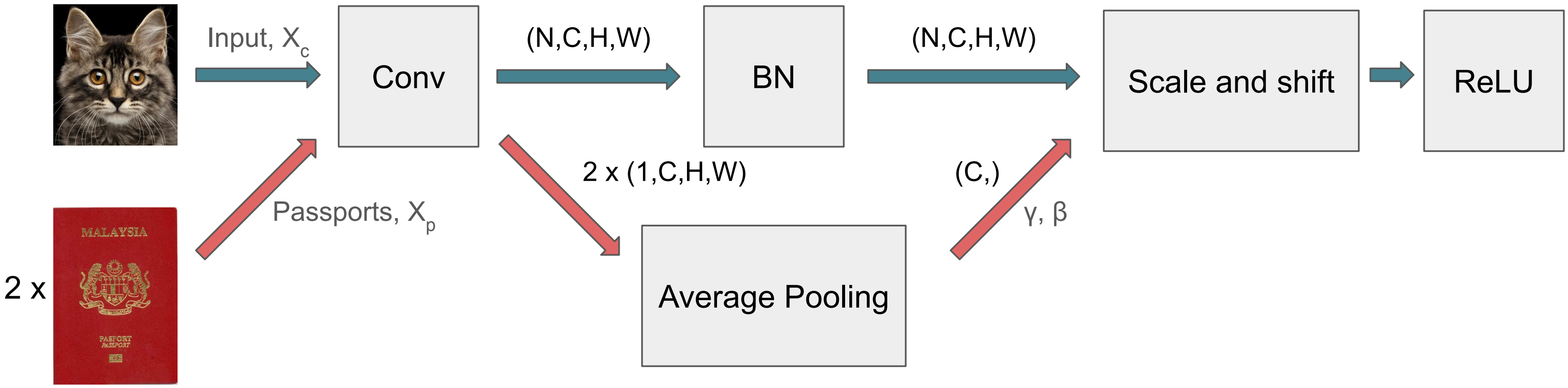


$$\mathbf{O}^l(\mathbf{X}_p) = \gamma^l \mathbf{X}_p^l + \beta^l = \gamma^l(\mathbf{W}_p^l * \mathbf{X}_c^l) + \beta^l, \tag{2}$$

$$\gamma^l = \mathrm{Avg}(\mathbf{W}_p^l * \mathbf{P}_\gamma^l), \qquad \beta^l = \mathrm{Avg}(\mathbf{W}_p^l * \mathbf{P}_\beta^l), \tag{3}$$

Figure at the left is the illustration of proposed passport layer in a ResNet block, where F is shown at equation 3 on figure above.

# Visualization of Passporting Layer



Input, $X_c$

Conv

(N,C,H,W)

BN

(N,C,H,W)

Scale and shift

ReLU

2 x

Passports, $X_p$

2 x (1,C,H,W)

Average Pooling

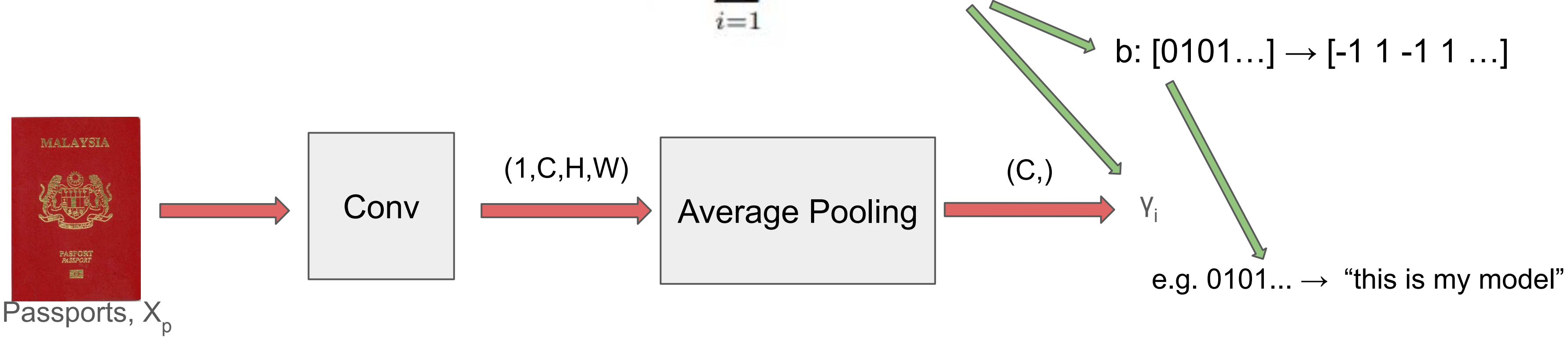
(C,)

$\gamma$, $\beta$

# Embedding Binary Signatures by Sign of Scale Factors

- During learning of the DNN weights, one can enforce scale factor to take either positive or negative signs as designated by adding the following **sign loss** regularization term into the objective function

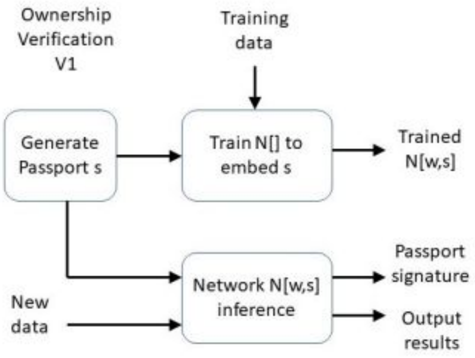
$$R(\gamma, \mathbf{P}, \mathbf{B}) = \sum_{i=1}^{C} \max(\gamma_0 - \gamma_i b_i, 0)$$



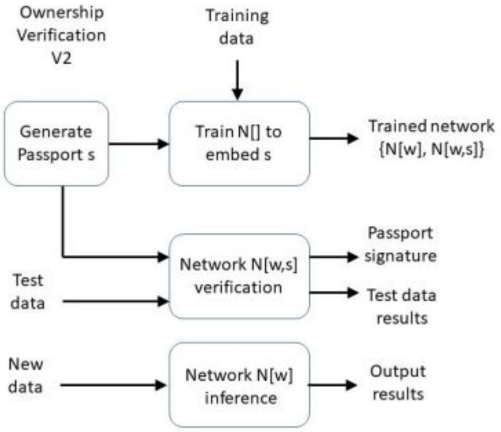
- It is robust due to **the sign of the scale factors cannot be switched after the model has converged.** Once the sign is switched, the model performance will degrade significantly due to the different output distribution of the passport layer.
- This feature **explains the superior robustness of embedded passports** against different ambiguity attacks.

# Ownership Verification with Passports

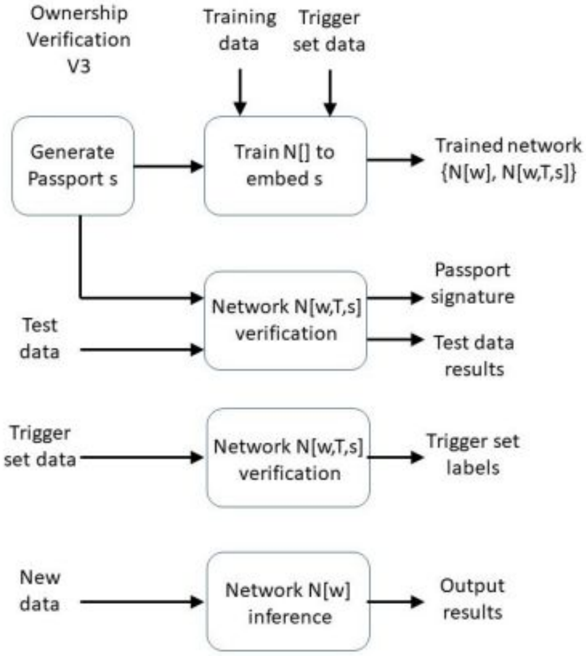
- Taking advantages of the proposed passport embedding method, we design three ownership verification schemes



Scheme V1



Scheme V2

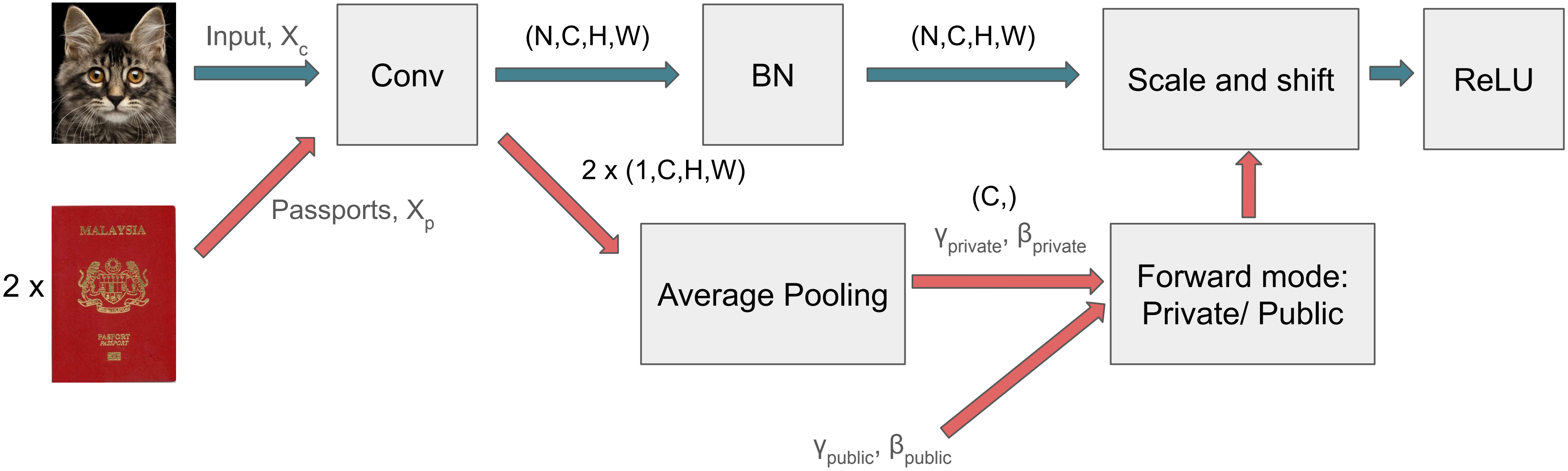


Scheme V3

# Ownership Verification with Passports

| Ownership verification schemes | Training | Inferencing | Verification |
|---|---|---|---|
| Scheme 1: Passport is distributed with the trained DNN model | - Minimize **target task** (i.e. classification) + regularization term (i.e. sign loss) | - Passports needed | - White-box mode<br>- Network ownership is verified by the distributed passports |
| Scheme 2: Private passport is embedded but not distributed | - Simultaneously achieve two goals<br>- Minimize **target task with no passporting layers** & **target task with passporting layer** + regularization term | - Do not need passports<br>- Network perform inference using $\gamma_{public}$, $\beta_{public}$ | - White-box mode<br>- Only carry out verification upon requested by the law enforcement, by adding passport layers, **detecting embedded sign signatures.** |
| Scheme 3: Both the private passport and trigger set are embedded but not distributed | - Training process is identical to scheme 2<br>- A **set of trigger images** is embedded in addition to the embedded passports [2] | - Inference process is identical to scheme 2 | - Black-box mode to claim ownership of the suspect DNN model through API calls.<br>- White-box mode verification is identical to scheme 2 |

# Visualization of Passporting Layer (Scheme 2 and 3)



Input, $X_c$

Conv

(N,C,H,W)

BN

(N,C,H,W)

Scale and shift

ReLU

2 x

Passports, $X_p$

2 x (1,C,H,W)

Average Pooling

(C,)

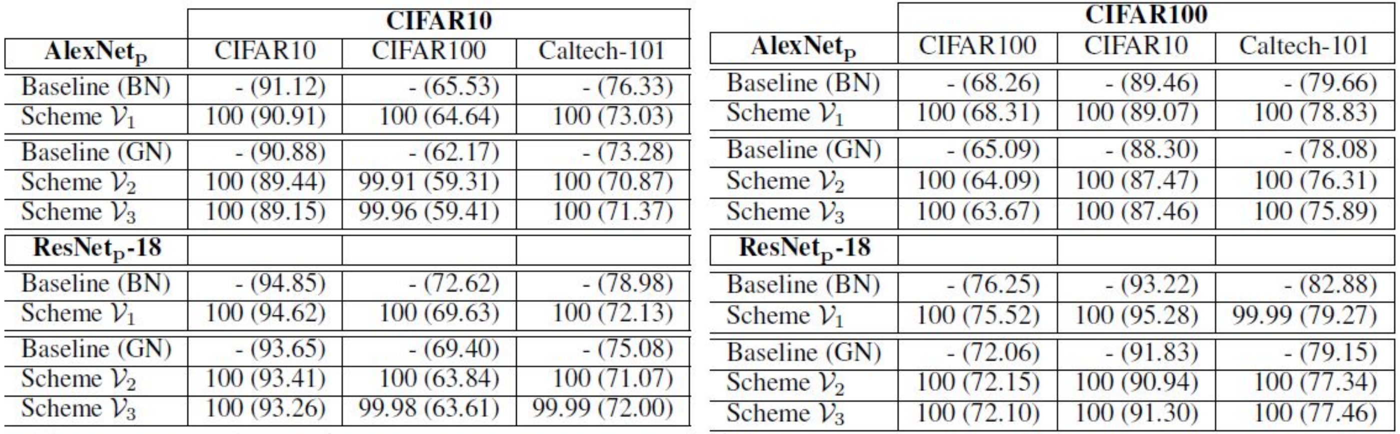
$\gamma_{private}, \beta_{private}$

Forward mode: Private/ Public

$\gamma_{public}, \beta_{public}$

# Experiment Results: Robustness against Removal Attack

Robustness against fine-tuning



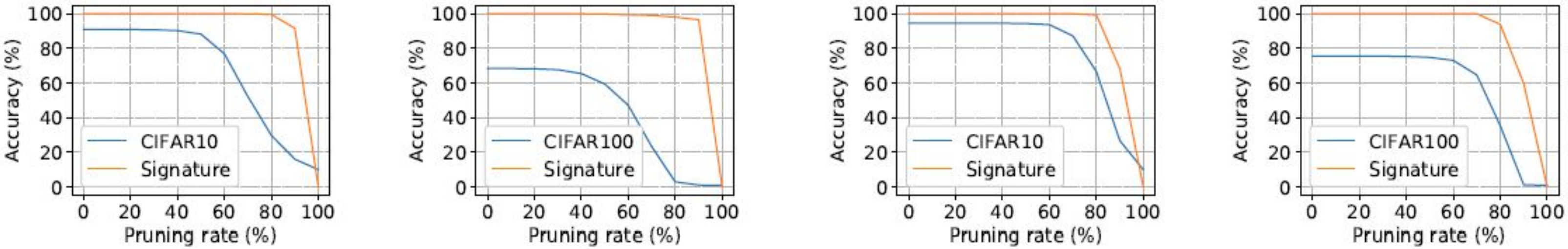
| | CIFAR10 | | |
|---|---|---|---|
| **AlexNet$_p$** | CIFAR10 | CIFAR100 | Caltech-101 |
| Baseline (BN) | - (91.12) | - (65.53) | - (76.33) |
| Scheme $\mathcal{V}_1$ | 100 (90.91) | 100 (64.64) | 100 (73.03) |
| Baseline (GN) | - (90.88) | - (62.17) | - (73.28) |
| Scheme $\mathcal{V}_2$ | 100 (89.44) | 99.91 (59.31) | 100 (70.87) |
| Scheme $\mathcal{V}_3$ | 100 (89.15) | 99.96 (59.41) | 100 (71.37) |
| **ResNet$_p$-18** | | | |
| Baseline (BN) | - (94.85) | - (72.62) | - (78.98) |
| Scheme $\mathcal{V}_1$ | 100 (94.62) | 100 (69.63) | 100 (72.13) |
| Baseline (GN) | - (93.65) | - (69.40) | - (75.08) |
| Scheme $\mathcal{V}_2$ | 100 (93.41) | 100 (63.84) | 100 (71.07) |
| Scheme $\mathcal{V}_3$ | 100 (93.26) | 99.98 (63.61) | 99.99 (72.00) |

| | CIFAR100 | | |
|---|---|---|---|
| **AlexNet$_p$** | CIFAR100 | CIFAR10 | Caltech-101 |
| Baseline (BN) | - (68.26) | - (89.46) | - (79.66) |
| Scheme $\mathcal{V}_1$ | 100 (68.31) | 100 (89.07) | 100 (78.83) |
| Baseline (GN) | - (65.09) | - (88.30) | - (78.08) |
| Scheme $\mathcal{V}_2$ | 100 (64.09) | 100 (87.47) | 100 (76.31) |
| Scheme $\mathcal{V}_3$ | 100 (63.67) | 100 (87.46) | 100 (75.89) |
| **ResNet$_p$-18** | | | |
| Baseline (BN) | - (76.25) | - (93.22) | - (82.88) |
| Scheme $\mathcal{V}_1$ | 100 (75.52) | 100 (95.28) | 99.99 (79.27) |
| Baseline (GN) | - (72.06) | - (91.83) | - (79.15) |
| Scheme $\mathcal{V}_2$ | 100 (72.15) | 100 (90.94) | 100 (77.34) |
| Scheme $\mathcal{V}_3$ | 100 (72.10) | 100 (91.30) | 100 (77.46) |

Table 2: Removal Attack (Fine-tuning): Detection/Classification accuracy (in %) of different passport networks where BN = batch normalisation and GN = group normalisation. (Left: trained with CIFAR10 and fine-tune for CIFAR100/Caltech-101. Right: trained with CIFAR100 and fine-tune for CIFAR10/Caltech-101.) Accuracy outside bracket is the signature detection rate, while in-bracket is the classification rate.

# Experiment Results: Robustness against Removal Attack

Robustness against pruning on AlexNet (Left) and ResNet18 (Right)

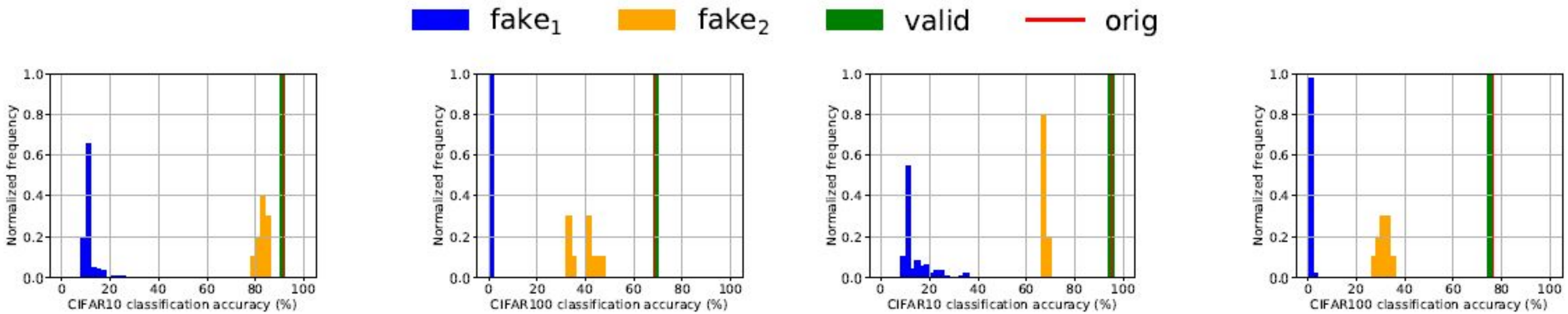


(a) $\text{AlexNet}_\mathbf{p}$ (b) $\text{ResNet}_\mathbf{p}$-18

Figure 3: Removal Attack (Model Pruning): Classification accuracy of our passport-based DNN models on CIFAR10/CIFAR100 and signature detection accuracy against different pruning rates.

# Experiment Results: Robustness against Ambiguity Attack

## Resilience against ambiguity attacks

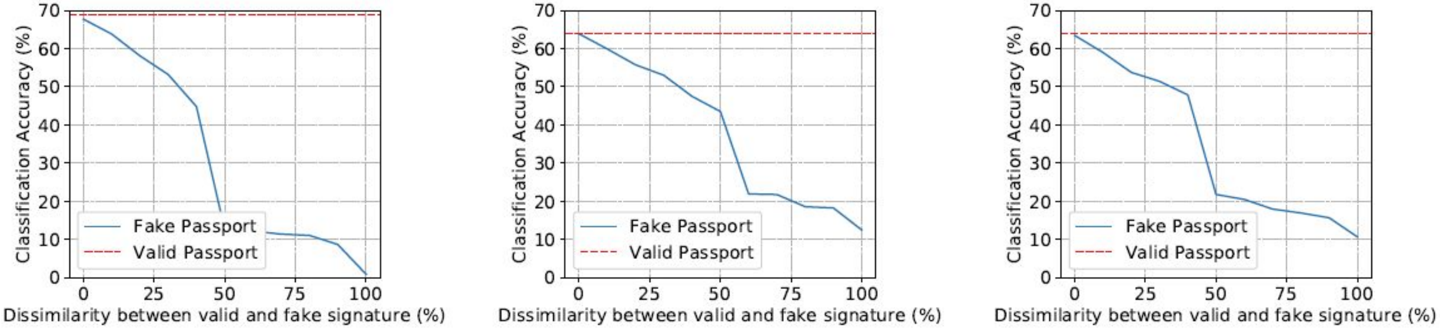


(a) $\text{AlexNet}_\text{p}$. (Left) CIFAR10, (Right) CIFAR100. (b) $\text{ResNet}_\text{p}$-18. (Left) CIFAR10, (Right) CIFAR100.

Figure 4: Ambiguity Attack: Classification accuracy of our passport networks with valid passport, *random attack* ($fake_1$) and *revered-engineering attack* ($fake_2$) on CIFAR10 and CIFAR100.

# Experiment Results: Robustness against Ambiguity Attack

## Resilience against ambiguity attacks



(a) Verification scheme $\mathcal{V}_1$

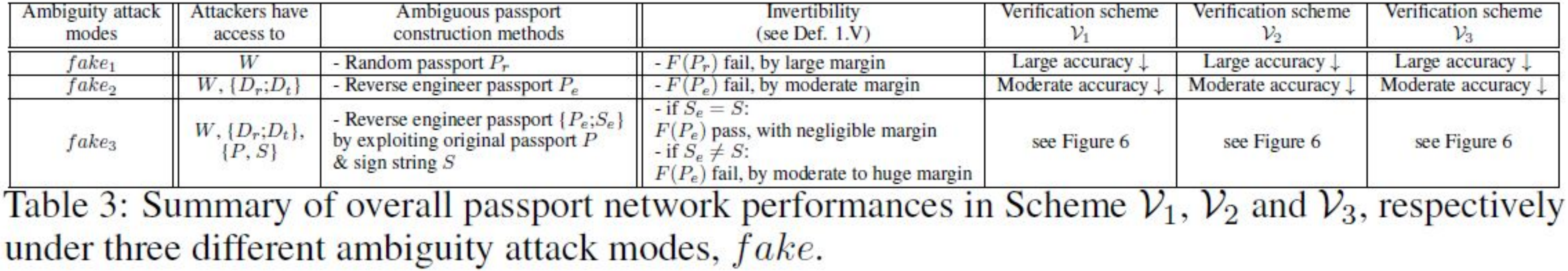
(b) Verification scheme $\mathcal{V}_2$

(c) Verification scheme $\mathcal{V}_3$

Figure 5: Ambiguity Attack: Classification accuracy on CIFAR100 under insider threat ($fake_3$) on three verification schemes. It is shown that when a correct signature is used, the classification accuracy is intact, while for a partial correct signature (sign scales are modified around 10%), the performance will immediately drop around 5%, and a totally wrong signature will obtain a meaningless accuracy (1-10%). Based on the threshold $\leq \epsilon_f = 3\%$ for AlexNet$_\mathrm{p}$ and by the fidelity evaluation process $F$, any potential ambiguity attacks (even with partially correct signature) are effectively defeated.

# Experiment Results: Summary

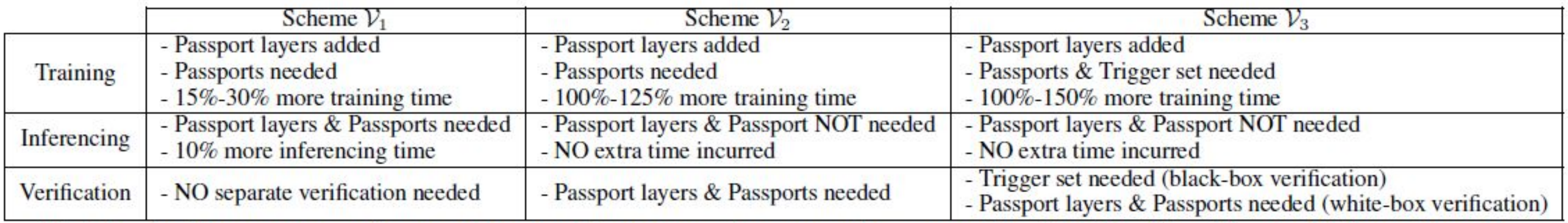
Summary of overall passport network performances



| Ambiguity attack modes | Attackers have access to | Ambiguous passport construction methods | Invertibility (see Def. 1.V) | Verification scheme $\mathcal{V}_1$ | Verification scheme $\mathcal{V}_2$ | Verification scheme $\mathcal{V}_3$ |
|---|---|---|---|---|---|---|
| $fake_1$ | $W$ | - Random passport $P_r$ | - $F(P_r)$ fail, by large margin | Large accuracy ↓ | Large accuracy ↓ | Large accuracy ↓ |
| $fake_2$ | $W$, $\{D_r;D_t\}$ | - Reverse engineer passport $P_e$ | - $F(P_e)$ fail, by moderate margin | Moderate accuracy ↓ | Moderate accuracy ↓ | Moderate accuracy ↓ |
| $fake_3$ | $W$, $\{D_r;D_t\}$,<br>$\{P, S\}$ | - Reverse engineer passport $\{P_e;S_e\}$ by exploiting original passport $P$ & sign string $S$ | - if $S_e = S$:<br>$F(P_e)$ pass, with negligible margin<br>- if $S_e \neq S$:<br>$F(P_e)$ fail, by moderate to huge margin | see Figure 6 | see Figure 6 | see Figure 6 |

Table 3: Summary of overall passport network performances in Scheme $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$, respectively under three different ambiguity attack modes, $fake$.

# Network Complexity



| | Scheme $\mathcal{V}_1$ | Scheme $\mathcal{V}_2$ | Scheme $\mathcal{V}_3$ |
|---|---|---|---|
| Training | - Passport layers added<br>- Passports needed<br>- 15%-30% more training time | - Passport layers added<br>- Passports needed<br>- 100%-125% more training time | - Passport layers added<br>- Passports & Trigger set needed<br>- 100%-150% more training time |
| Inferencing | - Passport layers & Passports needed<br>- 10% more inferencing time | - Passport layers & Passport NOT needed<br>- NO extra time incurred | - Passport layers & Passport NOT needed<br>- NO extra time incurred |
| Verification | - NO separate verification needed | - Passport layers & Passports needed | - Trigger set needed (black-box verification)<br>- Passport layers & Passports needed (white-box verification) |

Table 4: Summary of our proposed passport networks complexity for $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$ schemes.

# Conclusions

- We implemented **Passporting Layer** to solve problem on ambiguity attack and having superior robustness against removal attack and ambiguity attack.
- The DNN model with passport layers will **only perform well if and only if a valid passport is used**, else the performance will be significantly deteriorated.
- Even if an attacker is **able to forge a fake passport without deteriorating the model performance, the binary signature remains** in place. Hence, we can still verify the ownership of the model.
- If the forged binary signature is dissimilar from real signature, then the **performance will be significantly deteriorated.**

# Links

- Arxiv: https://arxiv.org/abs/1909.07830
- Code: https://github.com/kamwoh/DeepIPR

# References

[1] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin’ichi Satoh. Embedding watermarks into deep neural networks. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, pages 269–277, 2017

[2] Y Adi, C Baum, M Cisse, B Pinkas, and J Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In 27th USENIX Security Symposium (USENIX), 2018.

[3] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS), pages 159–172, 2018.