

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

Lixin Fan¹, Kam Woh Ng^{1,2}, Chee Seng Chan²

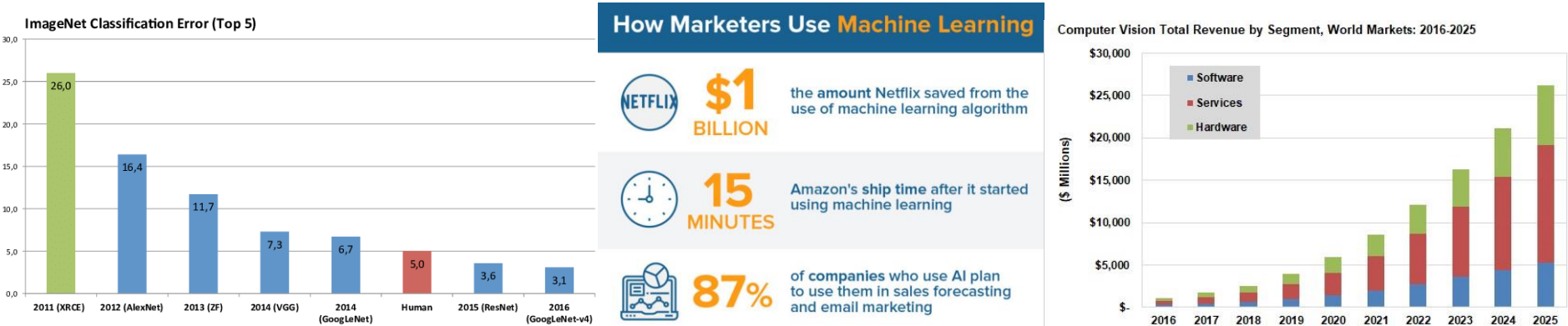
¹WeBank AI Lab, Shenzhen, China

²Center of Image and Signal Processing, University of Malaya,
Kuala Lumpur, Malaysia

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

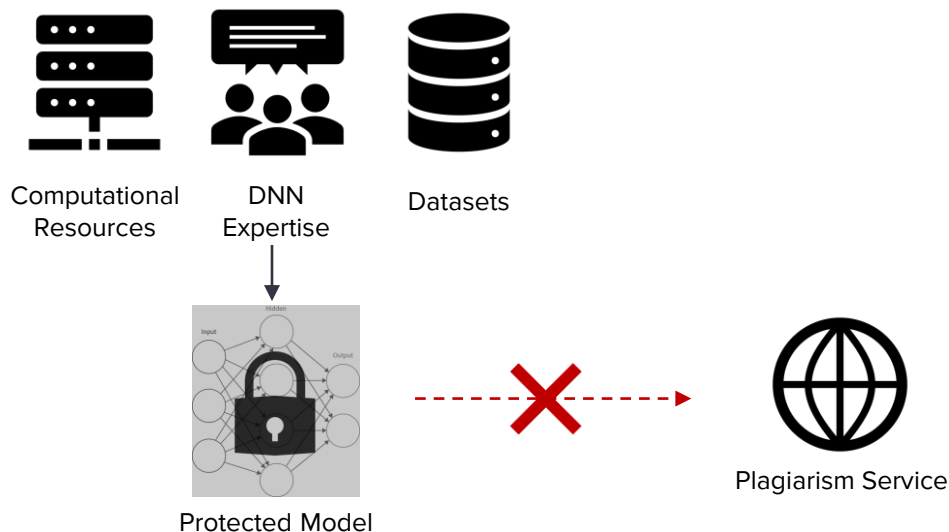
Machine Learning as a Service (MLaaS)

- More and more **business models and services** using Deep Neural Network (DNN)



Protection on DNN model is needed!

- Companies **invested a lot** to create powerful models
- They are **easily copied** and used by plagiarizers.
- We **need a protection** on DNN **from being illegally copied, distributed and abused.**



DNN watermarking methods

1. Feature based approach

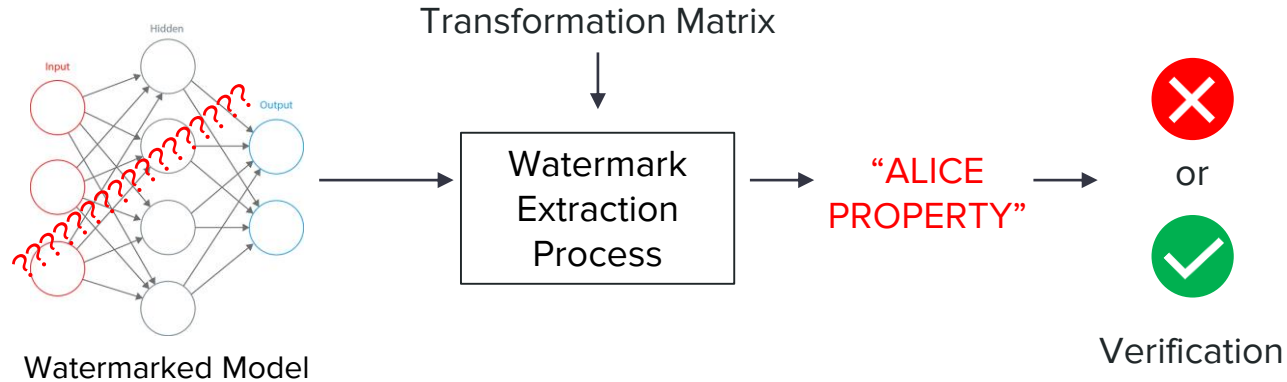
- Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, **“Embedding watermarks into deep neural networks”**
- B. D. Rouhani, H. Chen, and F. Koushanfar, **“Deepsigns: A generic watermarking framework for IP protection of deep learning models”**

2. Trigger-set based approach

- Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. **“Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring”**
- 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”**

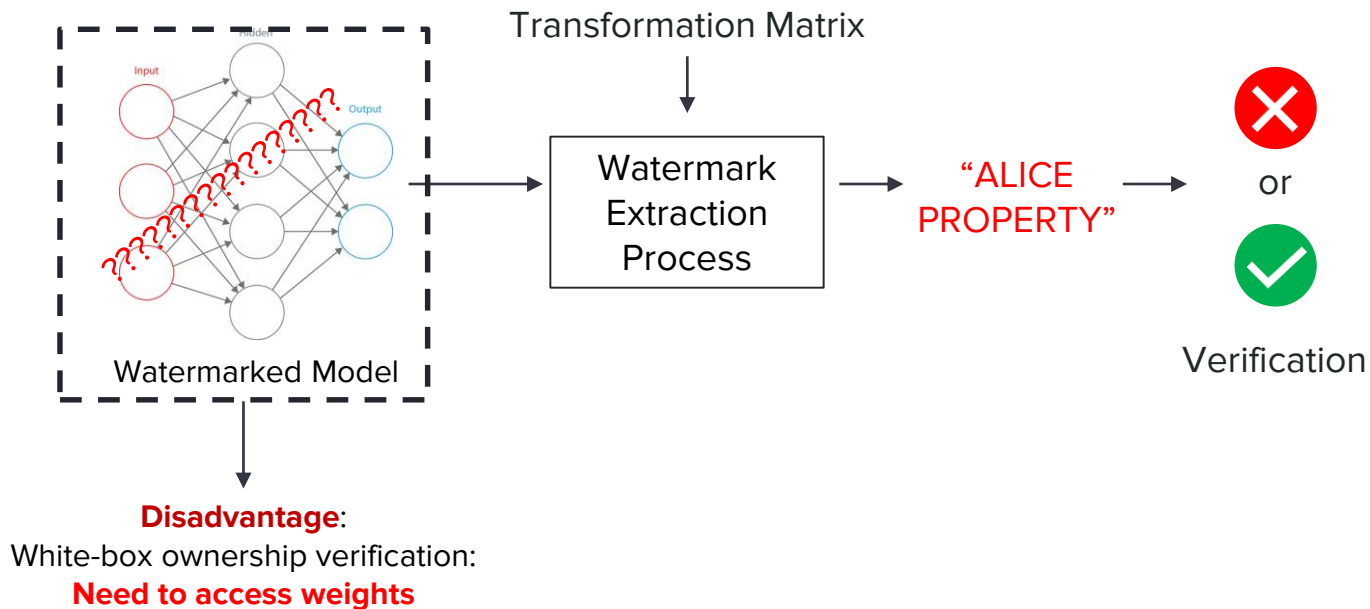
Feature-based approach

Feature based watermark detection



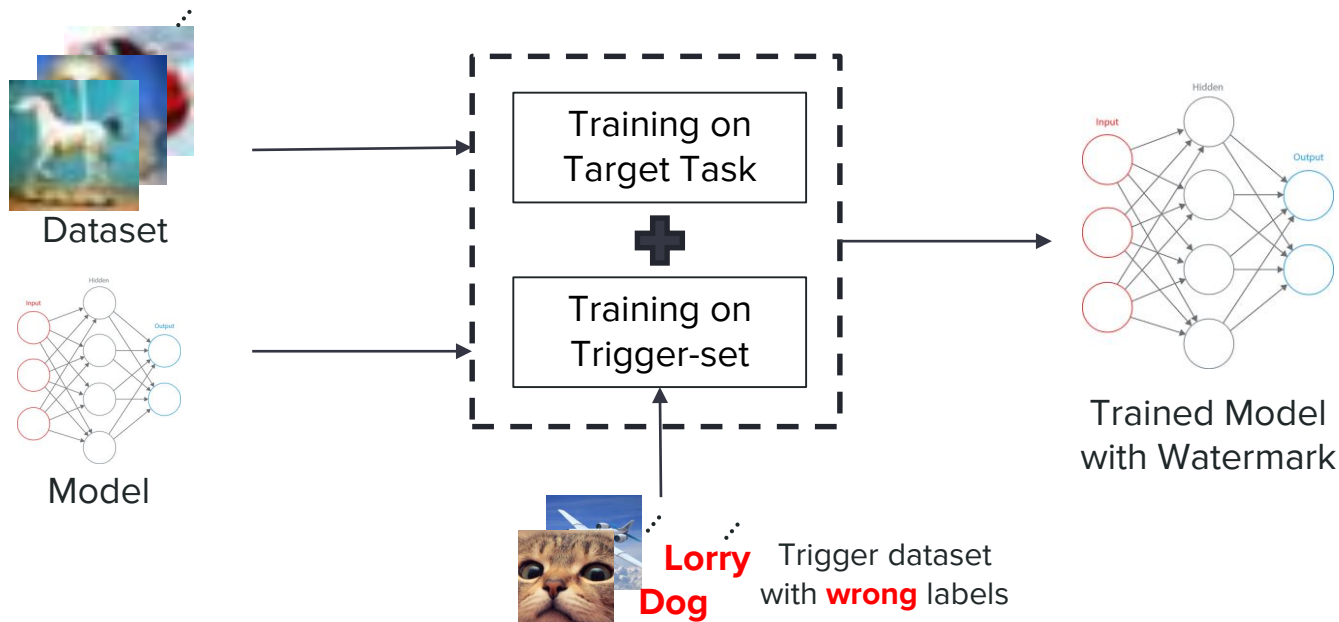
Feature-based approach

Feature based watermark detection



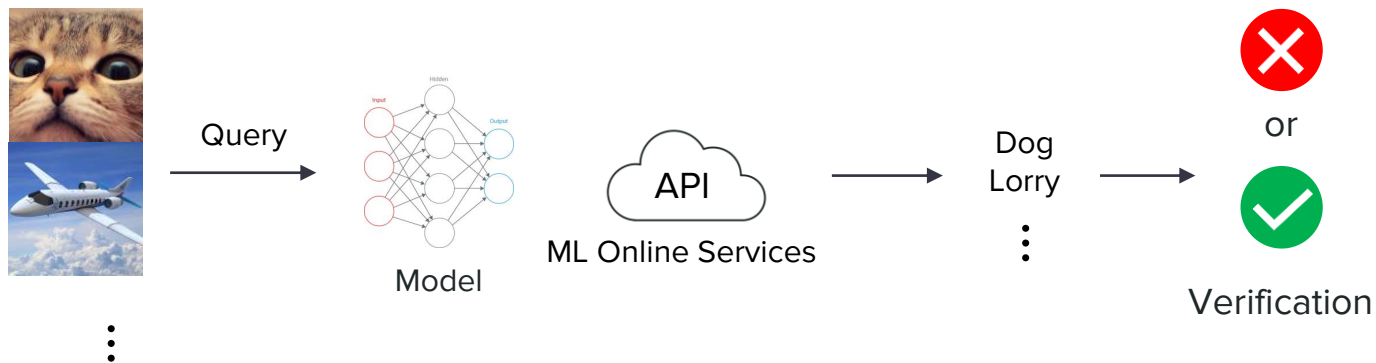
Trigger-set based approach

Trigger-set based watermark embedding



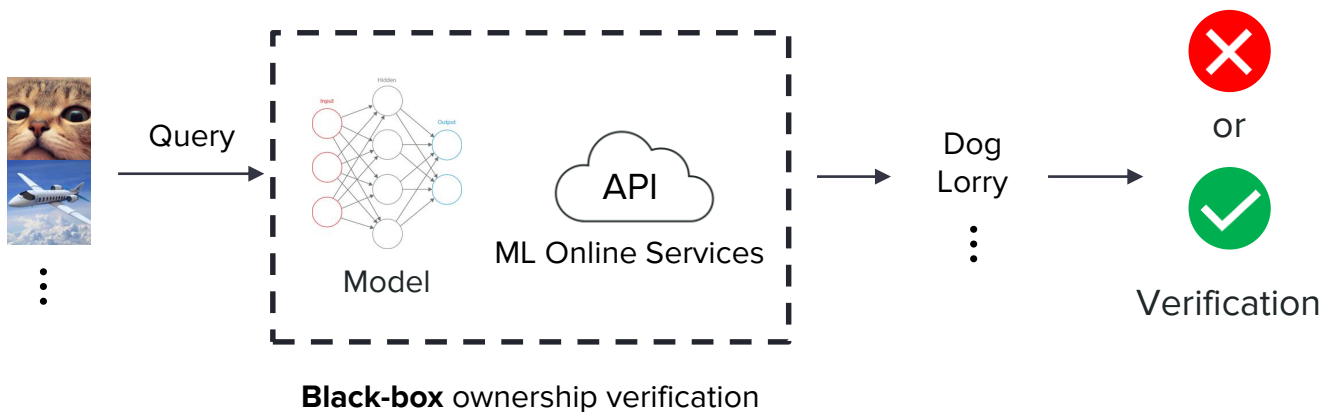
Trigger-set based approach

Trigger-set based watermark detection

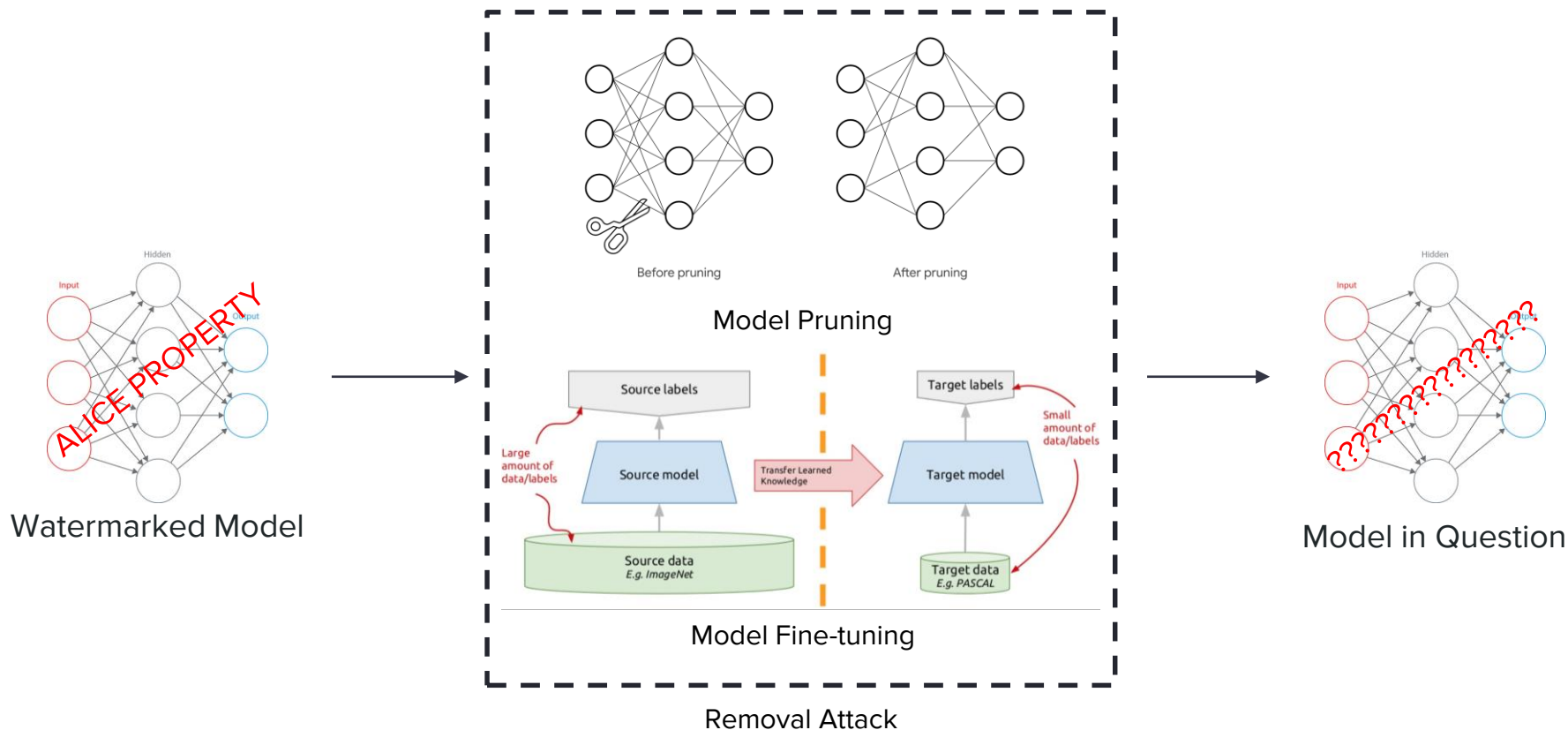


Trigger-set based approach

Trigger-set based watermark detection



Possible attacks to Ownership Protection



Effectiveness of Removal Attacks

- Watermark embedded in AlexNet for CIFAR10 classification

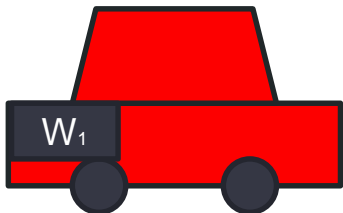
Removal Attacks	Feature based watermarking [1]	Trigger-set based watermarking [2]
Model Pruning	Strong (100% watermark detected with 65% pruning rate)	Strong (100% watermark detected with 70% pruning rate)
Fine-tuning (CIFAR10 → CIFAR100)	Strong (100% watermark detected after fine-tuning)	Weak (25% watermark detected after fine-tuning)

What is ambiguity attack?

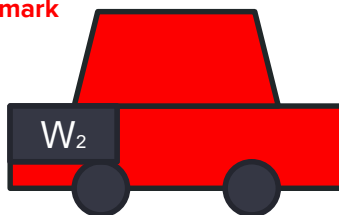
Alice bought a car



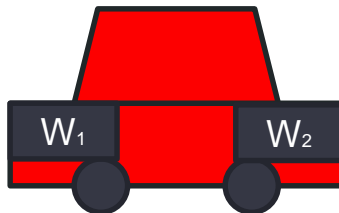
Alice **spray watermark on her car**



Bob stole Alice's car,
replace the watermark



Or he **spray watermark on the other side**

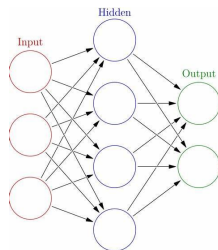


Police man does not know who is guilty by looking at the watermark

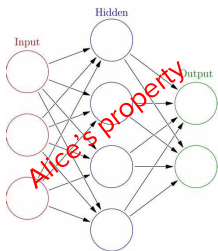


Ambiguity Attack to DNN models

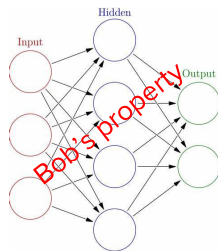
Alice trained a DNN model



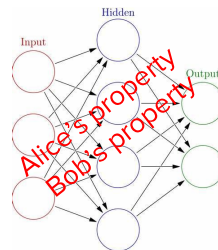
Alice **embeds watermark** into her model



Bob stole Alice's model,
replace the watermark



Or he **embeds his fake watermark** into the model



Judger confused due to two different watermarks are being detected from the model



Effectiveness of Ambiguity Attack

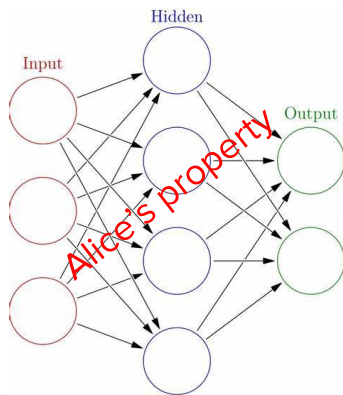
- Watermark embedded in AlexNet for CIFAR10 classification

Watermark approach	Real Watermark	Fake Watermark
Feature based (White-box)	100% watermark detected	100% watermark detected
Trigger-set based (Black-box)	100% watermark detected	100% watermark detected

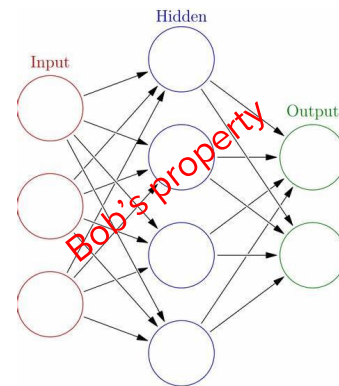
Watermark detection rate for both **real** and **fake** watermarks

How to deal with **ambiguity** attack?

Current Situation

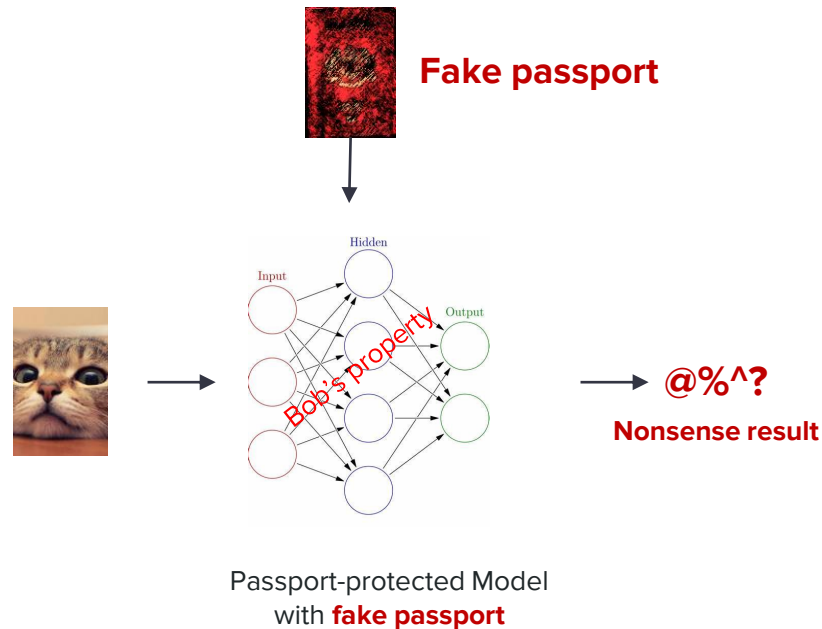
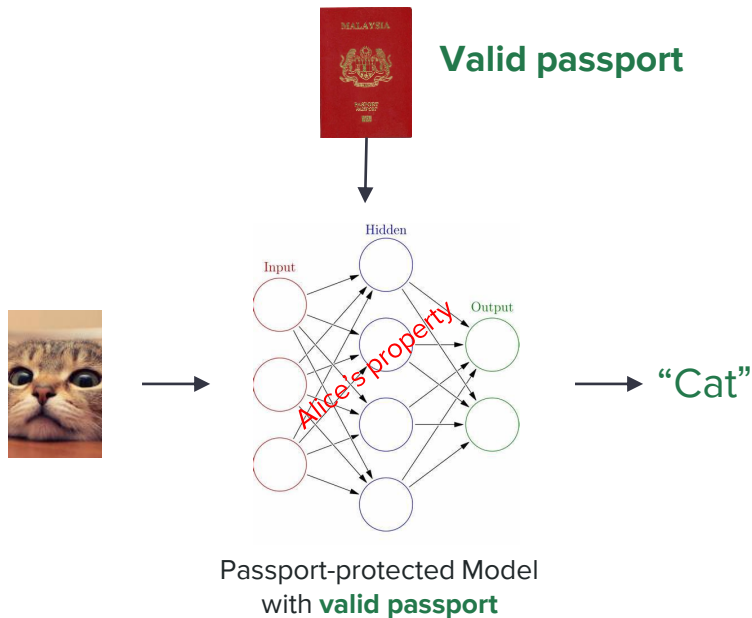


Passport-protected Model
with original watermark



Copied Model
with fake watermark

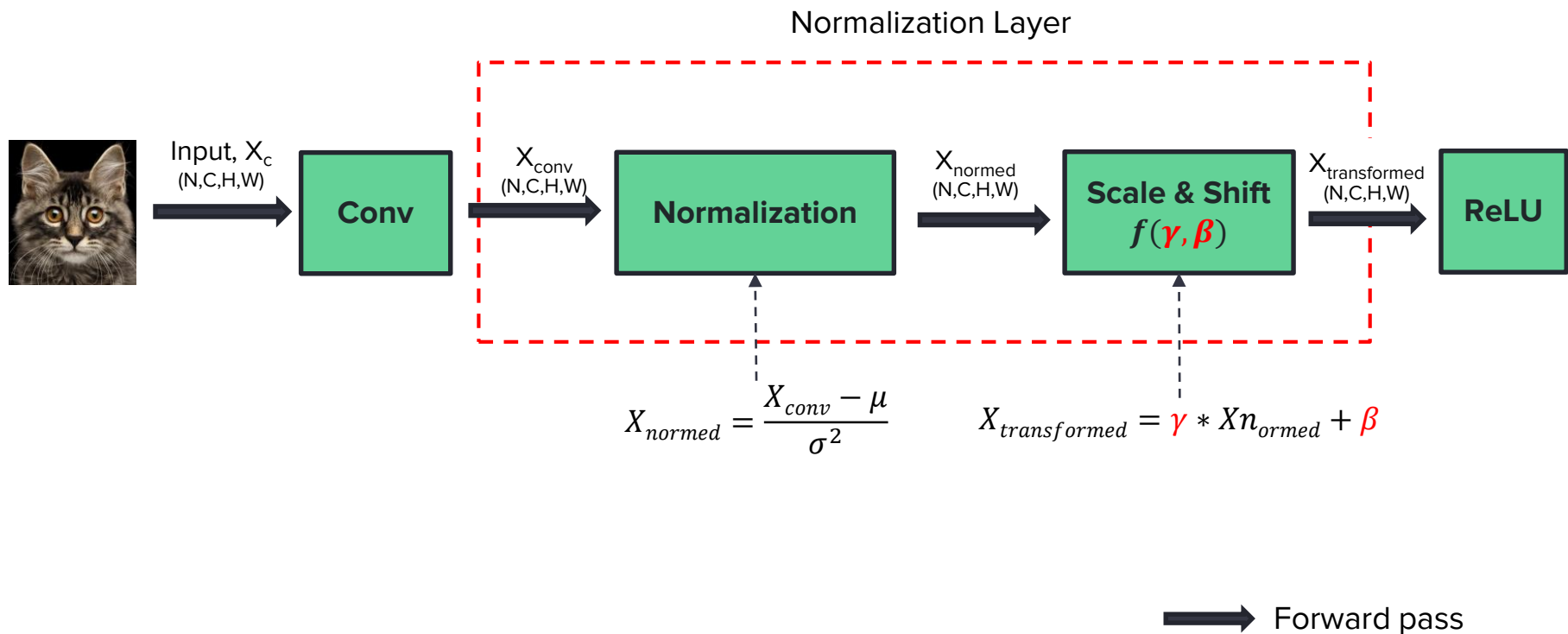
Proposed Solution



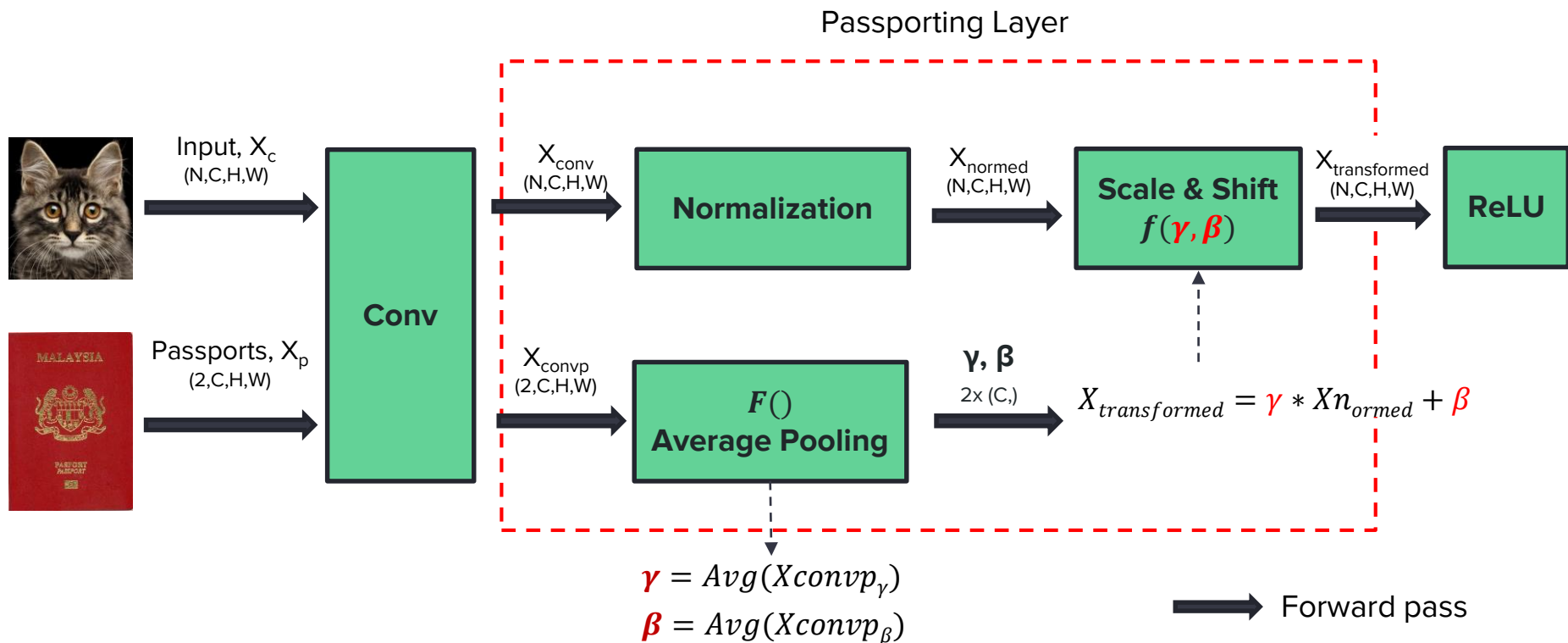
Aim:

- Model cannot function without valid passport

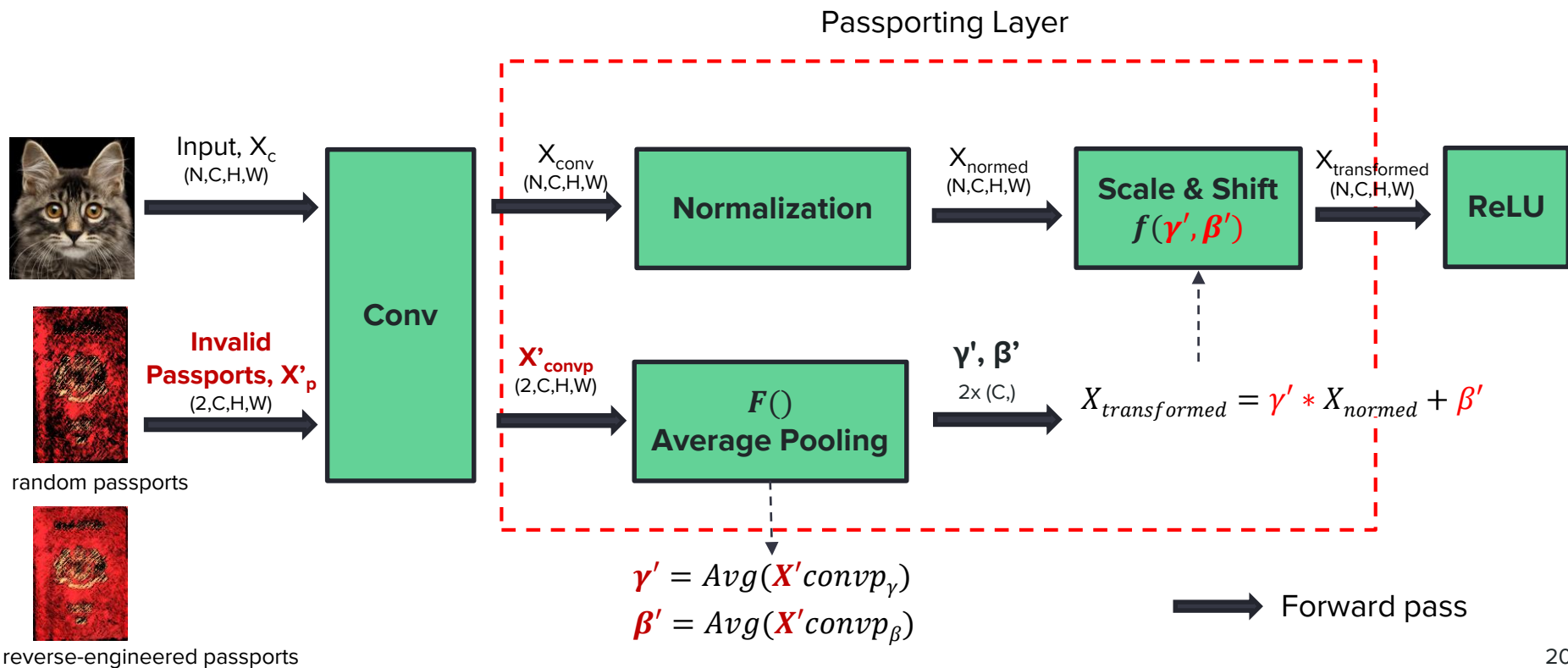
Conventional Convolution Layer



Passporting Layer





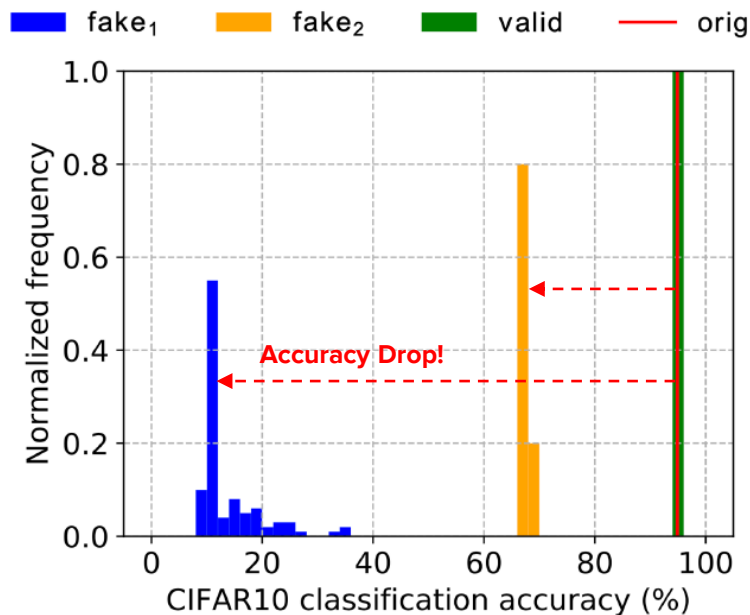
Passporting Layer



Effectiveness of Passport Protection

Result of Invalid passports

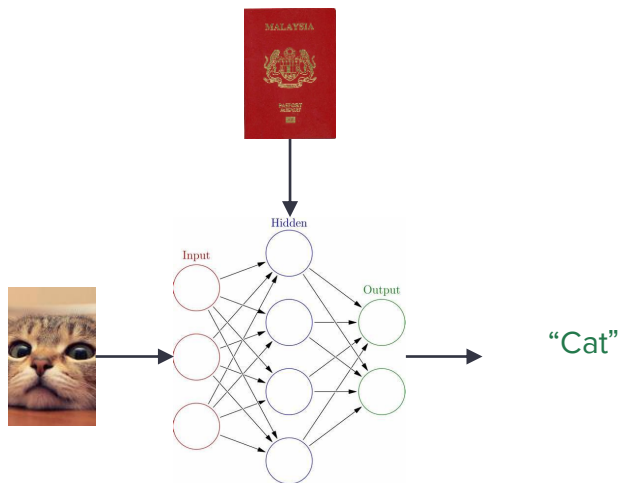
Ambiguity attack	Effect
Fake ₁ (random passports) 	Random guessing (at max 35%)
Fake ₂ (reverse-engineered passports) 	Performance deteriorated (at max 70%)



Example of ResNet_p-18 performance on CIFAR10 when performing different ambiguity attacks (fake₁ & fake₂)

Ownership Verification with Passports (Scheme 1)

Training & Inference



Passport is distributed with the trained DNN model

Model ownership is verified by **passports, performance and signature**

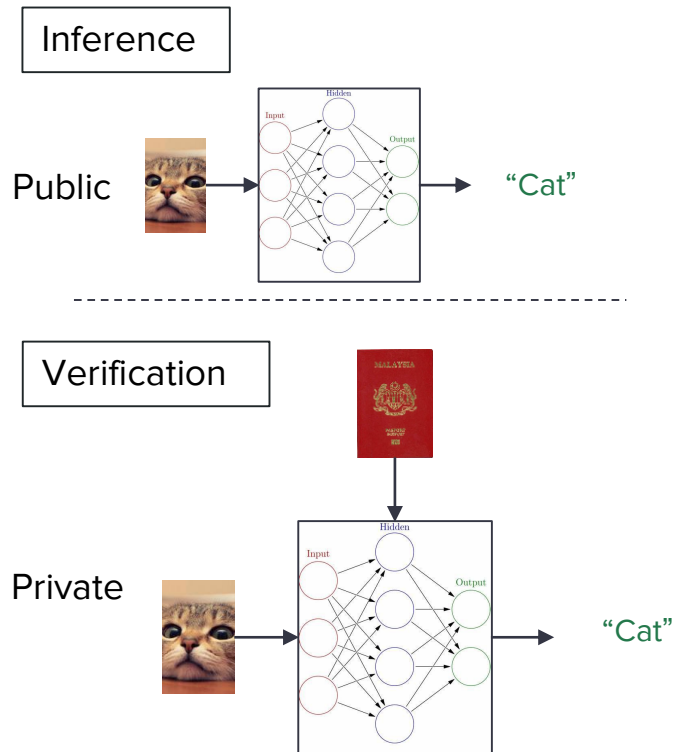
Verification type:

- **White-box**

Disadvantage:

1. **Need to distribute the passports**
2. **Extra inference time**

Ownership Verification with Passports (Scheme 2)



Private passport is embedded but not distributed

Verification type:

- **White-box**

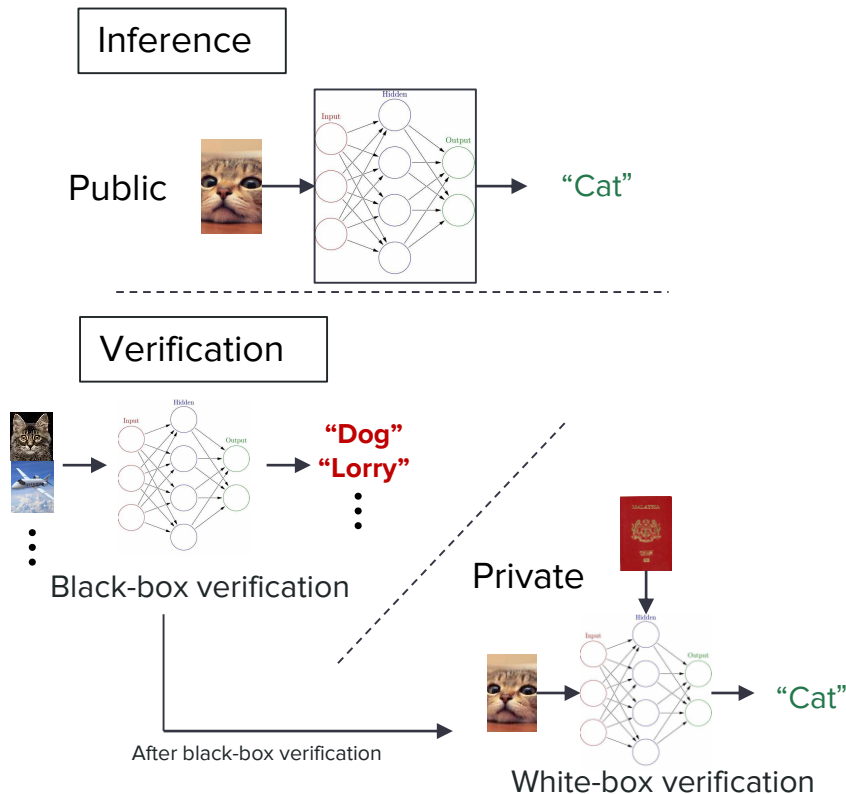
Advantage:

1. **No need to distribute the passports**
2. **No extra inference time**

Disadvantage:

- **Still a white-box verification**

Ownership Verification with Passports (Scheme 3)



Both the private passport and trigger set are embedded but not distributed

Black-box model ownership is verified by **query API calls**

Verification type:

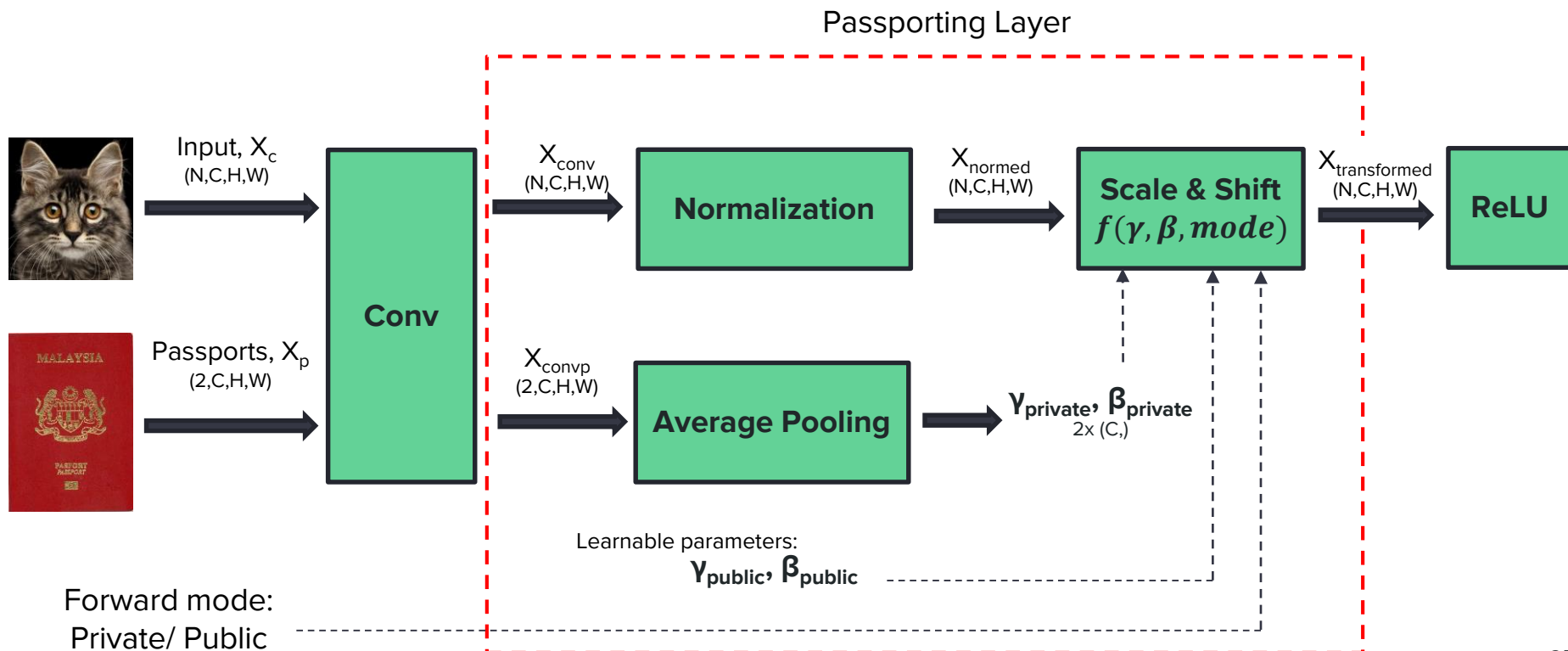
1. **Black-box**
2. **White-box**

Disadvantages of Scheme 1 & 2 Solved:

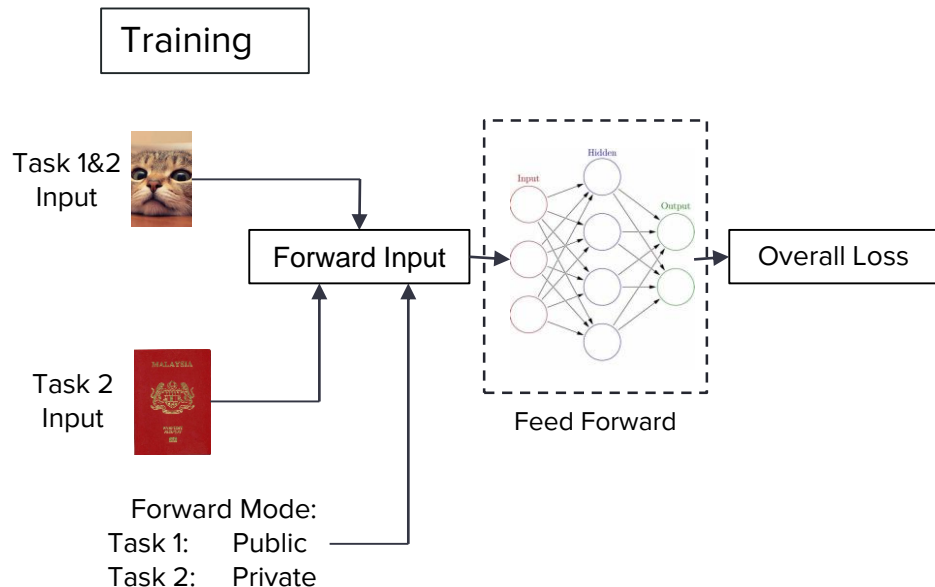
1. **No need to distribute the passports**
2. **No extra inference time**
3. **Able to have an initial suspect through black-box verification**

Passporting Layer (Scheme 2 & 3)

➡ Forward pass



Training on Scheme 2 & 3



Multi-task training:

Simultaneously minimizing following cost functions:

Scheme 2

Task 1: Cross Entropy (Public)

Task 2: Cross Entropy + Sign Loss (Private)

Overall Loss: $L = L_{\text{Task 1}} + L_{\text{Task 2}}$

Scheme 3

Task 1: Cross Entropy (Public)

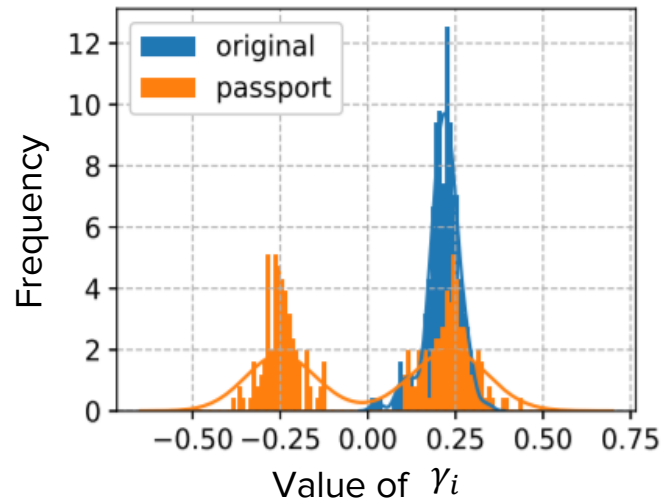
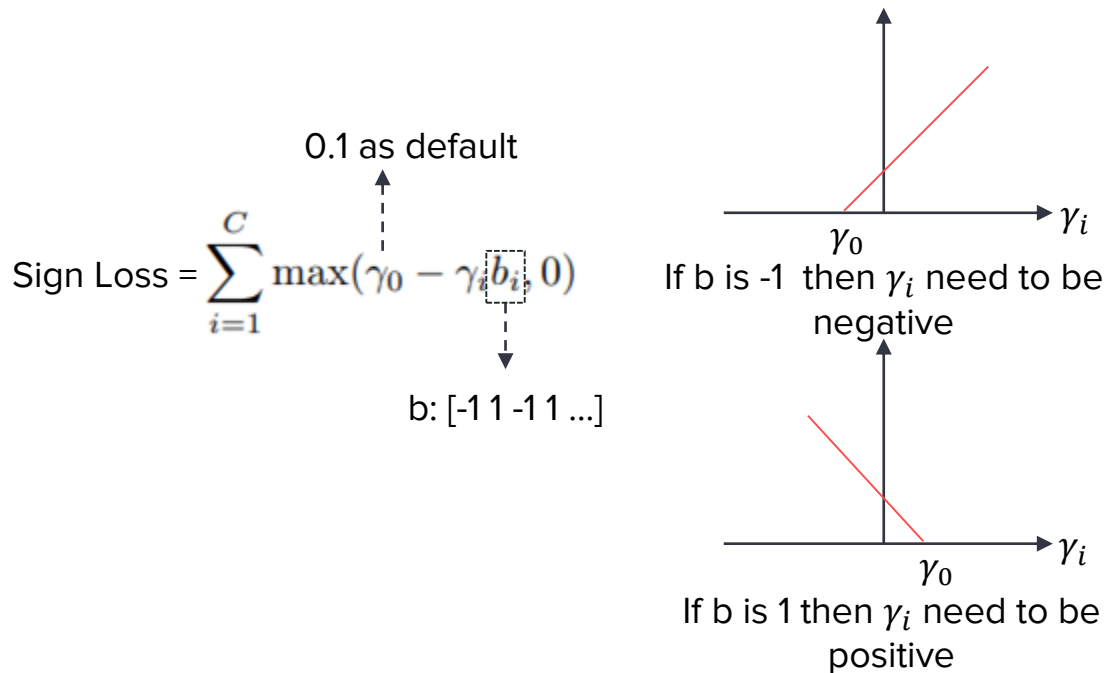
Task 2: Cross Entropy + Sign Loss (Private)

Task 3: Trigger-set Embedding Loss (Public + Private)

Overall Loss : $L = L_{\text{Step 1}} + L_{\text{Step 2}} + L_{\text{Step 3}}$

Embedding Binary Signatures by Sign of Scale Factors

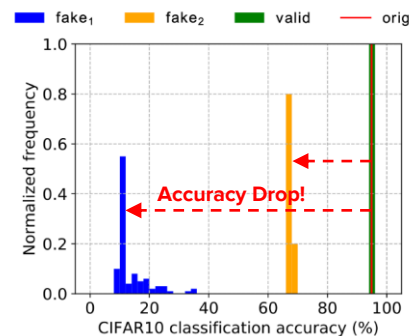
- **Enforce scale factor** to take **either positive or negative signs** as designated
 - Using hinge-loss like of regularization: **Sign-Loss**
 - **64 channels can embed 8 bytes signature**






Summary of Ambiguity Attacks

Summarized result done on: AlexNet & ResNet18

Datasets: CIFAR10 & CIFAR100



Ambiguity Attacks	Inference Phase	Verification Phase
 <p>Fake₁, Random Passport</p>	<ul style="list-style-type: none"> - Random Guessing - Useless Model 	<ul style="list-style-type: none"> - Useless Infringement
 <p>Fake₂, Reverse-Engineered Passport</p>	<ul style="list-style-type: none"> - Deteriorated Performance - Useless Model 	<ul style="list-style-type: none"> - Useless Infringement
 <p>Fake₃, Copied Passport</p>	<ul style="list-style-type: none"> - Performance Remained - Signature Detected 	<ul style="list-style-type: none"> - Ownership Verified

Summary of Ownership Verification Schemes

	Scheme 1	Scheme 2	Scheme 3
Need to distribute passport	Yes	No	No
Inference time	Up to 10%** more time	No extra time	No extra time
Training time	Up to 30%** more time	Up to 150%** more time	Up to 150%** more time
Black or White box Verification	White	White	Black & White

**Time increases are linearly depending on complexity of the network architecture

Take Home message

- **Protection on DNN** is urgently needed!
- **Existing** watermarking approaches are **vulnerable to ambiguity attack**
- **Passport-based approach** provided better protection in terms of **robustness against removal attack and ambiguity attack**
- **Passport-protected DNN model** will **only perform well if and only if a valid passport is used**, else 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.