Aliasing Backdoor Attacks on Pre-trained Models

Cheng'an Wei[1,2], Yeonjoon Lee[3], Kai Chen*[1,2], Guozhu Meng[1,2], and Peizhuo Lv[1,2]

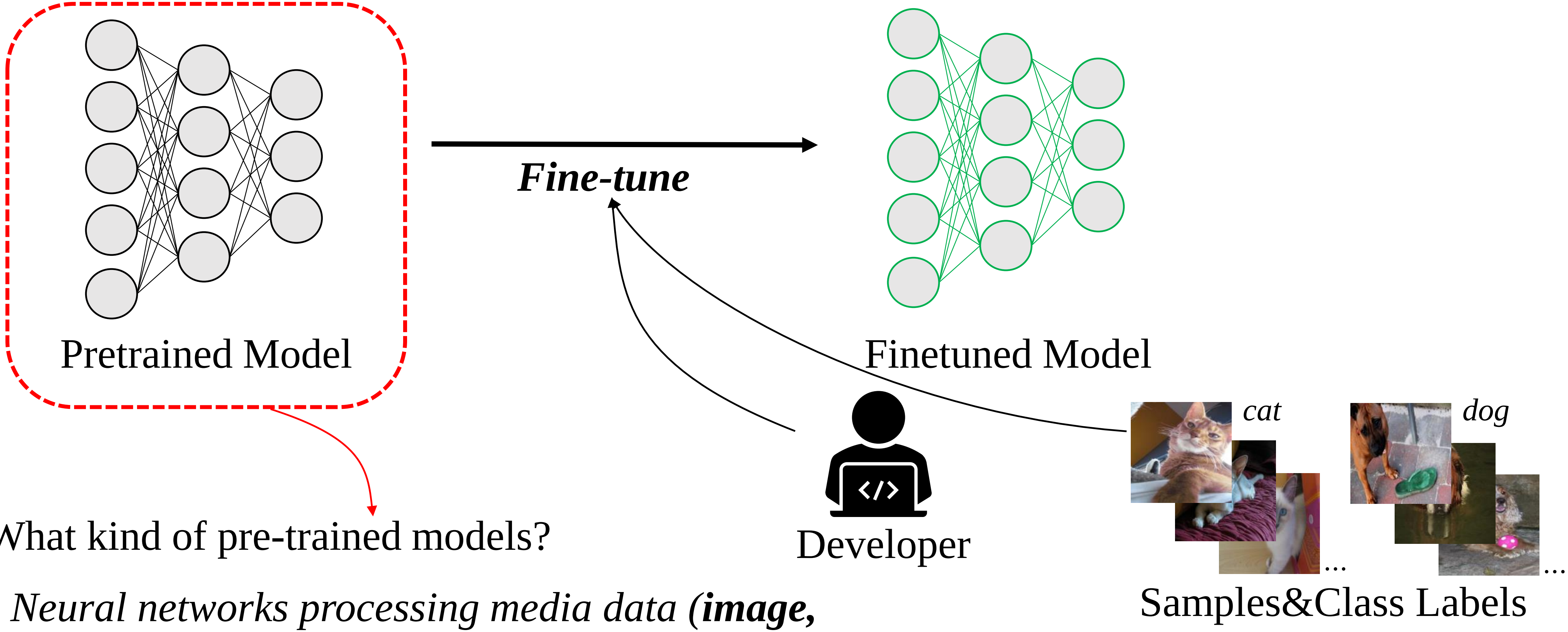
[1] *SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences, China*

[2] *School of Cyber Security, University of Chinese Academy of Sciences, China*

[3] *Hanyang University, Ansan, Republic of Korea*

{weichengan, chenkai, mengguozhu, lvpeizhuo}@iie.ac.cn, yeonjoonlee@hanyang.ac.kr

Background: Pre-trained Models in Transfer Learning



What kind of pre-trained models?

- *Neural networks processing media data (**image, audio**) e.g., computer vision, speech recognition*

- *Neural networks with **strided layers inside***

Background: Strided Layers in Pre-trained Models

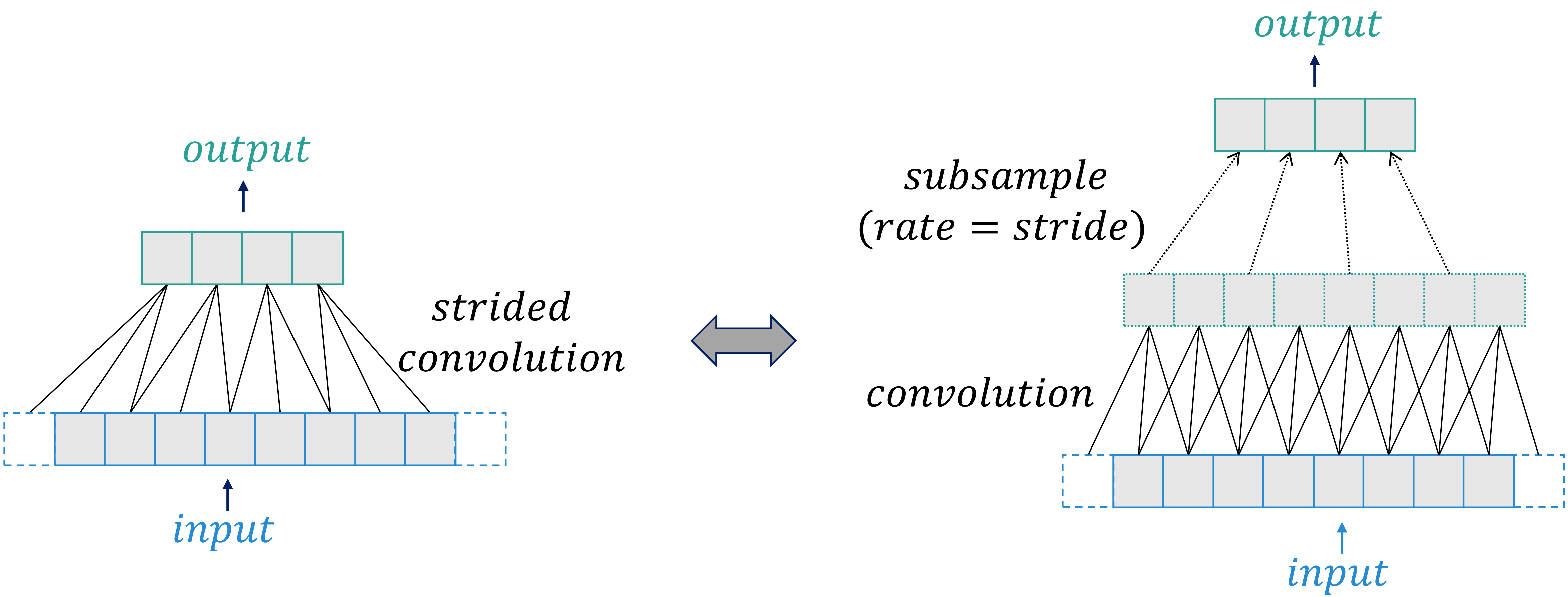
Strided layers: convolutional layer of stride ≥ 2

- *They are widely used in mainstream networks, e.g., ResNet and ViT.*
- *They are typically deployed as the first layer.*

```
>>> resnet50
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), pad
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
  (act1): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dila
  (layer1): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=Tr
      (act1): ReLU(inplace=True)
```

```
>>> vit_base
VisionTransformer(
  (patch_embed): PatchEmbed(
    (proj): Conv2d(3, 768, kernel_size=(16, 16), stride=(16, 16))
    (norm): Identity()
  )
  (pos_drop): Dropout(p=0.0, inplace=False)
  (norm_pre): Identity()
  (blocks): Sequential(
    (0): Block(
      (norm1): LayerNorm((768,), eps=1e-06, elementwise_affine=True)
      (attn): Attention(
```

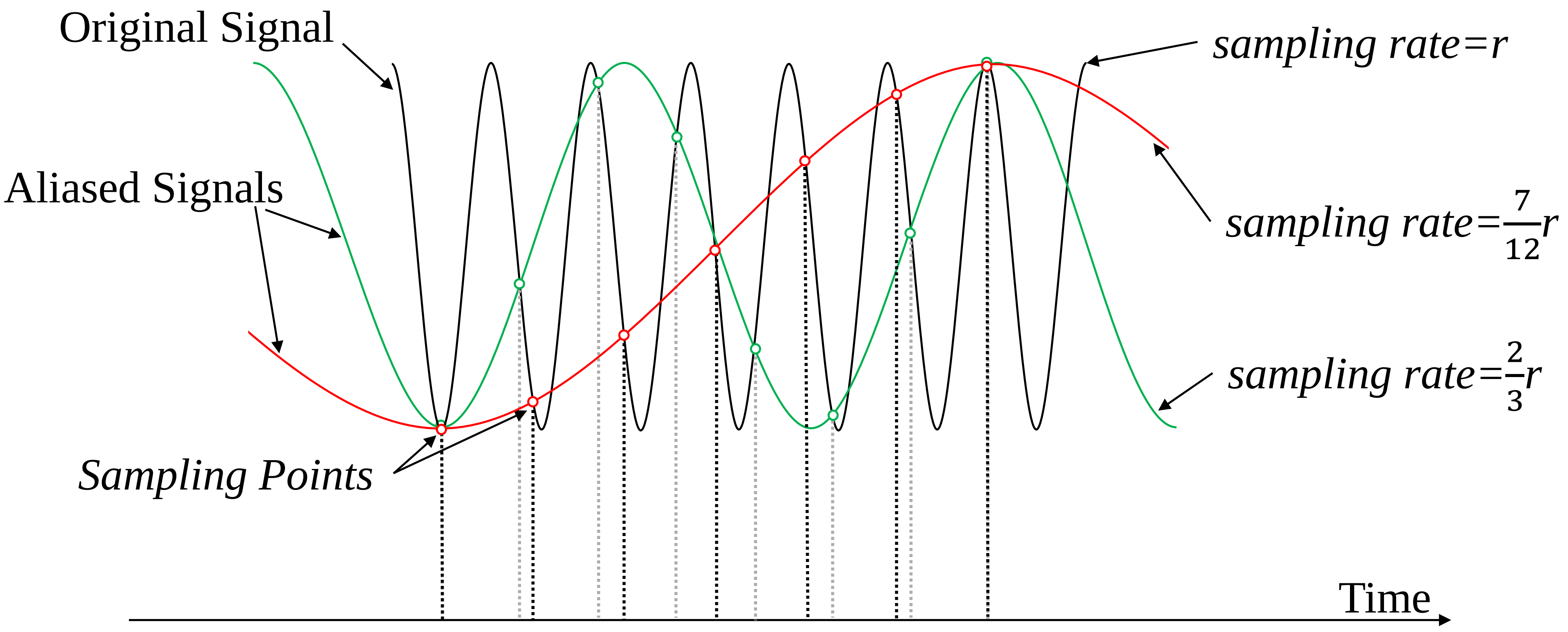
Observation: Subsampling in strided Layers



Observation 1: A strided layer involves a subsampling operation implicitly.

Observation: Aliasing Effect of Subsampling

Subsampling can result in aliasing effect:

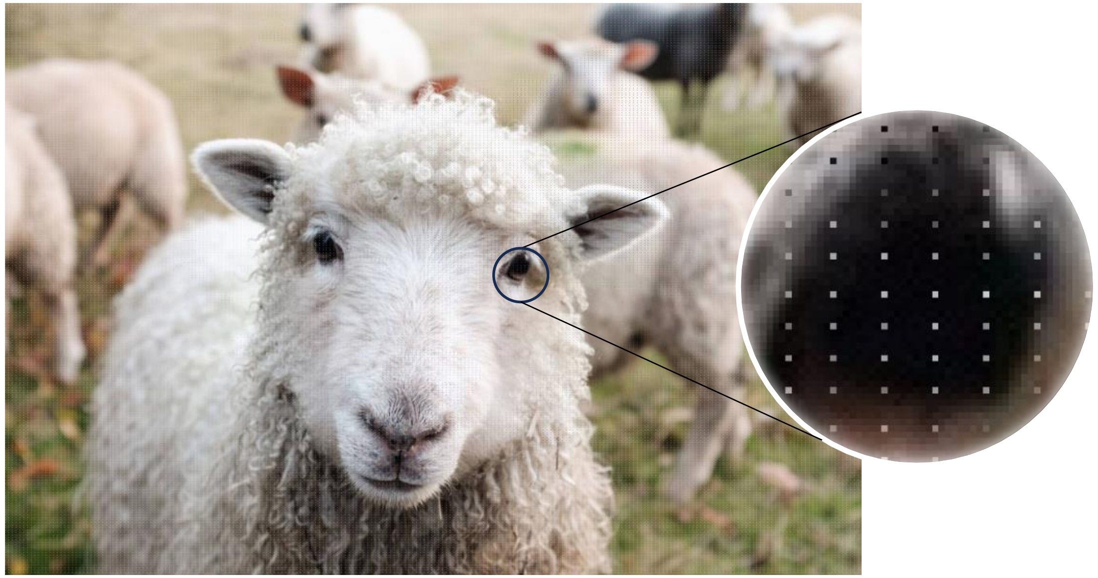


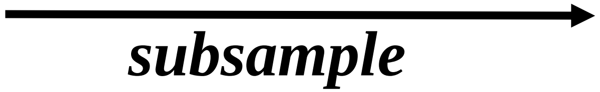
The aliased signal can be manipulated by perturbing the sampling points.

Observation: Aliasing Effect of Subsampling

The aliased signal can be manipulated by perturbing the sampling points:

e.g., Image-scaling attack (USENIX Security ‘19)

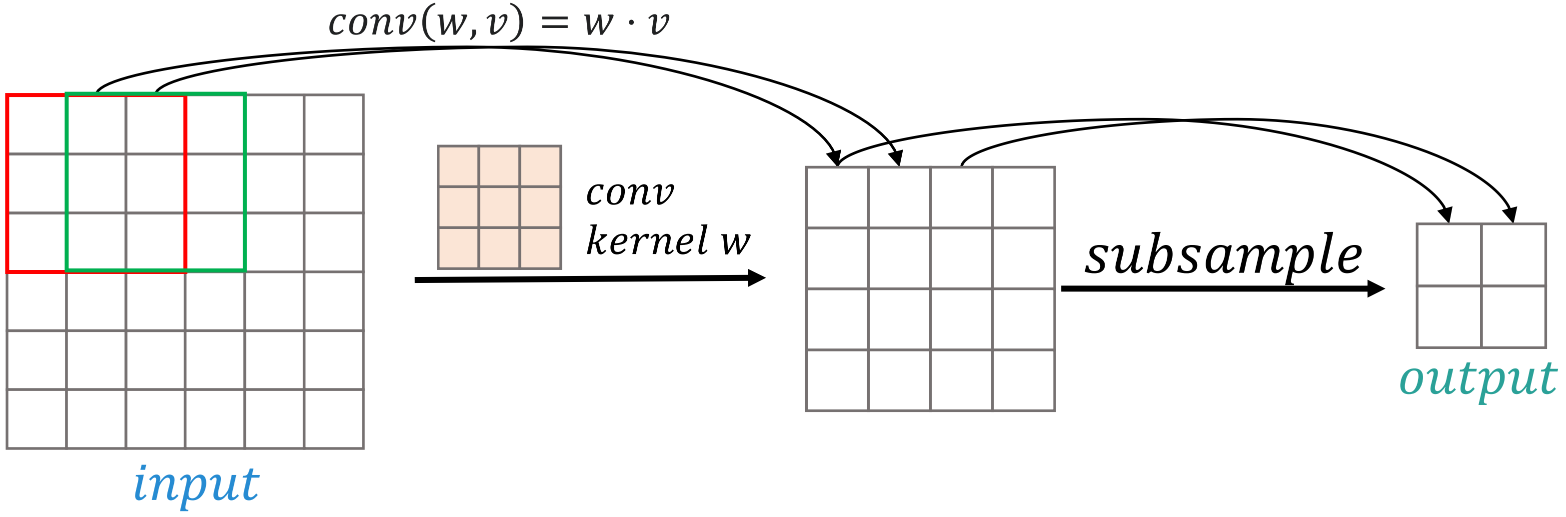




Observation 2: subsampling → aliasing → manipulation attack

Motivation: Aliasing for Backdoor Attack

A strided layer:



So…

Observation 1: A strided layer involves a subsampling operation implicitly.

Observation 2: subsampling → aliasing → manipulation attack

⬇

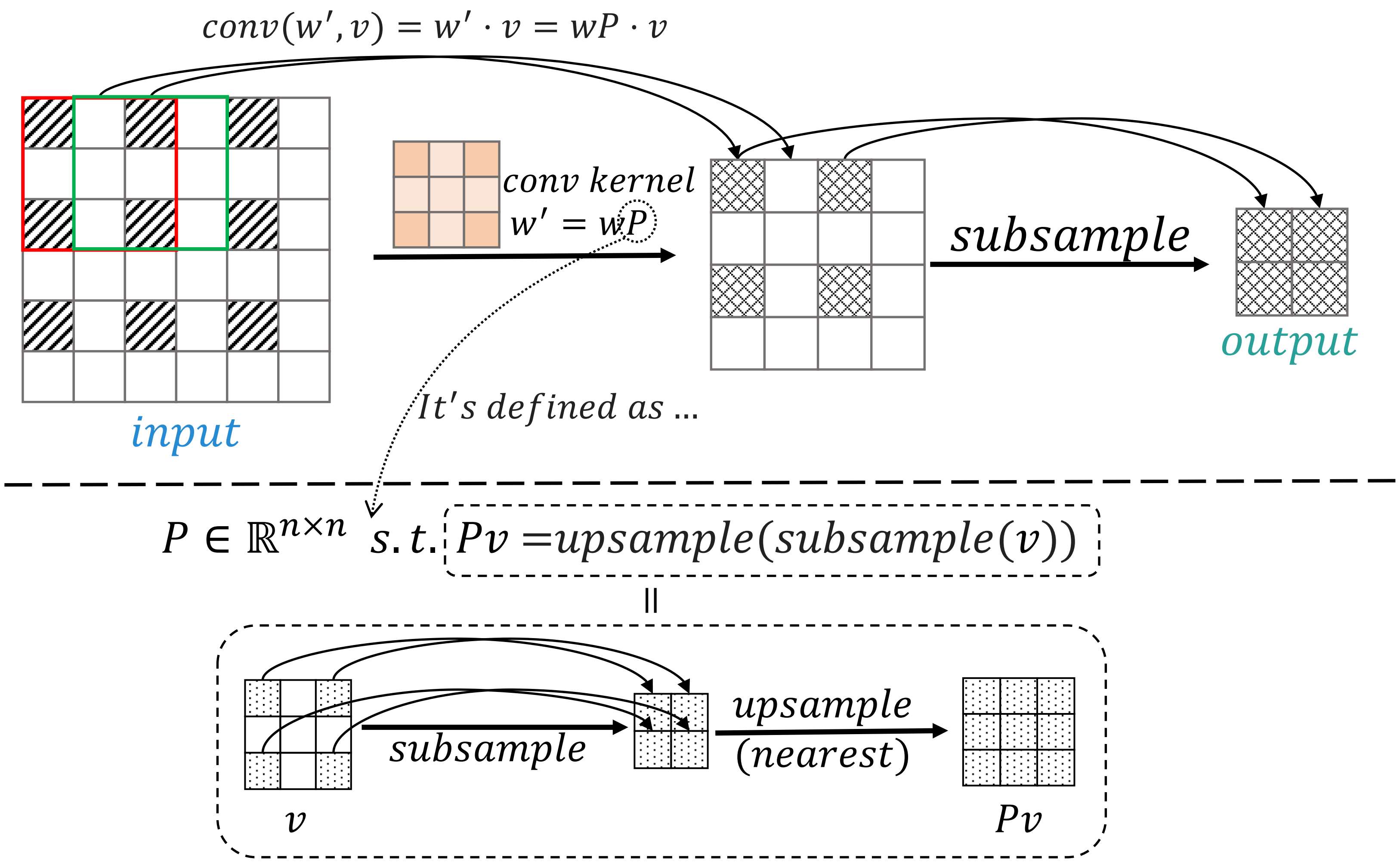
Q: Can we manipulate the output of a strided layer by exploiting aliasing?

A: Yes. By a backdoor attack.

Motivation: Aliasing for Backdoor Attack

An example of manipulating the output of a strided layer by

Creating aliasing intentionally by a modified convolution kernel.



Behave like a backdoor:

- *manipulated input → targeted aliasing → attacker-specified output*
- *benign input → non-sense noise → normal output*

Method: Aliasing for Backdoor Attack

How to launch aliasing backdoor attack on a pre-trained model?

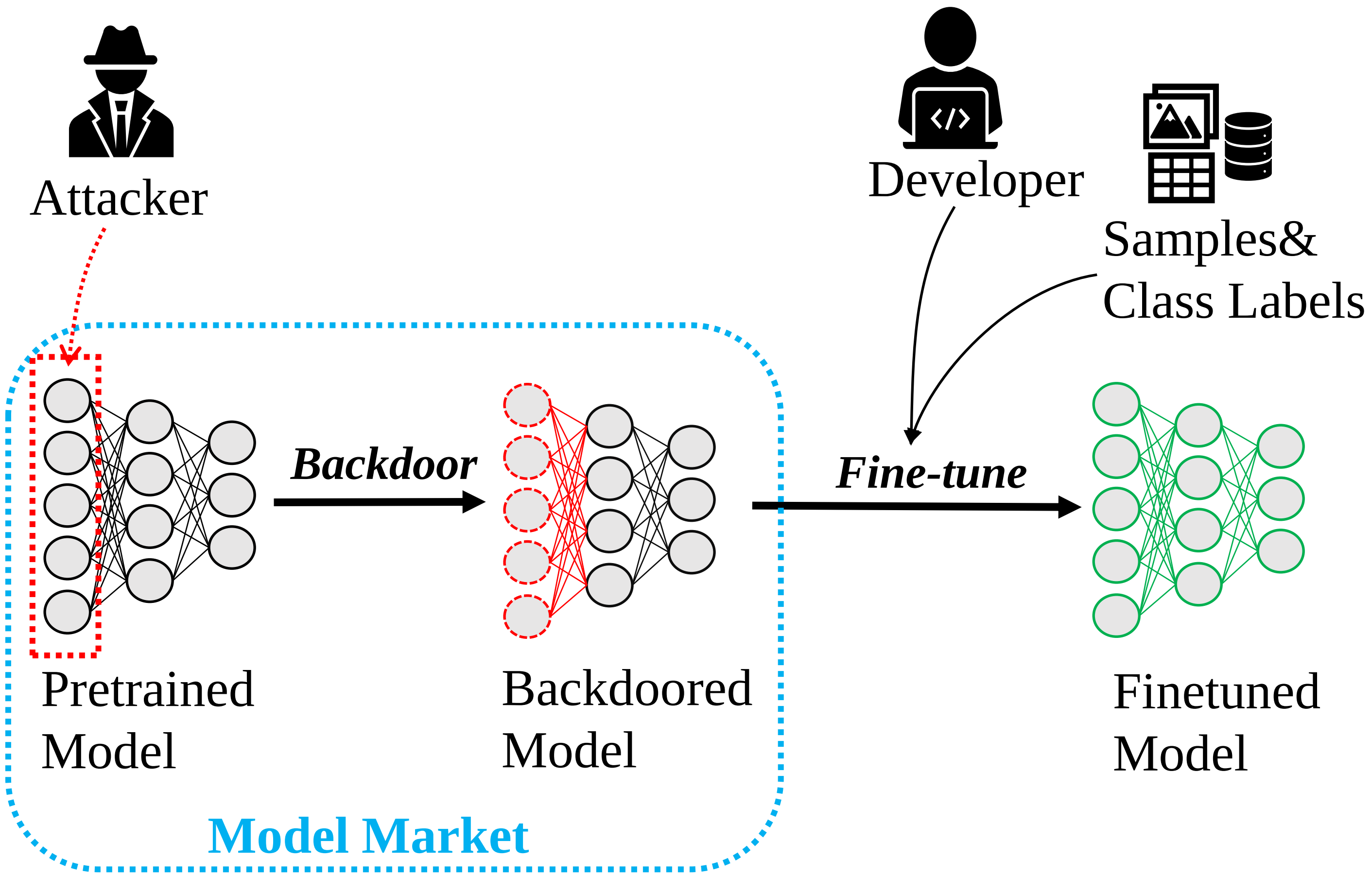
Backdoor Insertion:

- *Model* $\rightarrow$ *strided layer* $\hat{f}$
- *convolution kernel* w^k $\rightarrow$ *Matrix* P^k
- *Weight perturbation:* $w^k = w^k P$

The victim fine-tune the model…

Generate triggers for inputs:

- *Input* x_s *and target label* C $\rightarrow$ *trigger* δ
- $x_s + \delta \rightarrow$ *attacker-specified output*



Method: Adaptive Backdoor Insertion

How to launch aliasing backdoor attack on a pre-trained model?

Backdoor Insertion:

- *Model* $\rightarrow$ *strided layer* $\hat{f}$
- *convolution kernel* w^k $\rightarrow$ *Matrix* P^k
- *Weight perturbation:* $w^k = w^k P$

The victim fine-tune the model…

Generate triggers for inputs:

- *Input* x_s *and target label* C $\rightarrow$ *trigger* δ
- $x_s + \delta \rightarrow$ *attacker-specified output*

How to do this?

We search for matrix P^k *adaptively for different types of strided layers of different strides, kernels…*

Considerations:

- *Attack effectiveness*
- *Model utility*

Method: Adaptive Backdoor Insertion

No re-training or data required, completed in minutes and transfer to all downstream models.

Search matrix P with two different inputs x_1, x_2

s.t.:

1. *a manipulation attack is caused*
   - *minimize layer output difference*
   - *minimize input perturbation*
2. *the negative impact is minimized*
   - *minimize distance to the identity matrix*
   - *P is from a constrained space*

$$\min_{P^1,\cdots,P^K,x} \|\hat{f}'(x) - \hat{f}(x_1)\|_2 + \beta_1 \|\phi(x) - \phi(x_2)\|_2 + \beta_2 \sum_{k=1}^{K} \|P^k - I\|_2$$

$$\text{s.t.} \quad x = clamp(x), \quad w^k_{\hat{f}'} = w^k_{\hat{f}} \cdot P^k, \quad P^k \in \mathcal{C}(\mathbb{R}^{n\times n})$$

Constrained Space: $P \in \mathcal{C}(\mathbb{R}^{n\times n})$

constraint 1 : $\sum_j P_{i,j} = 1$

constraint 2 : $P_{i,j} = 0, if\ j \notin O(v_i)$

v_1	v_2	v_3
v_4	v_5	v_6
v_7	v_8	v_9

image case: $O(v_5) = \{2,4,5,6,8\}$

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9

audio case: $O(v_5) = \{4,5,6\}$

Method: Dynamic Trigger Generation

How to launch aliasing backdoor attack on a pre-trained model?

Backdoor Insertion:

- *Model* $\rightarrow$ *strided layer* $\hat{f}$
- *convolution kernel* w^k $\rightarrow$ *Matrix* P^k
- *Weight perturbation:* $w^k = w^k P$

The victim fine-tune the model…

Generate triggers for inputs:

- *Input* x_s *and target label* C $\rightarrow$ *trigger* δ
- $x_s + \delta \rightarrow$ *attacker-specified output*

How to do this?

Generate δ *by a target sample* x_t *from target label:*

layer output difference

$$\min_{\delta} \|\hat{f}'(x_s+\delta) - \hat{f}'(x_t)\|_2 + \lambda \cdot \|\phi(x_s+\delta) - \phi(x_s)\|_2$$

input perturbation strength

$$\text{s.t.} \quad \delta = clamp(x_s+\delta) - x_s$$

highly similar feature $\rightarrow$ *same prediction*

Method: Dynamic Trigger Generation

How to launch aliasing backdoor attac

Backdoor Insertion:

- *Model $\rightarrow$ strided layer $\hat{f}$*
- *convolution kernel w^k*
  $\rightarrow$ Matrix P^k
- *Weight perturbation: $w^k = w^k P$*

The victim fine-tune the model…

Generate triggers for inputs:

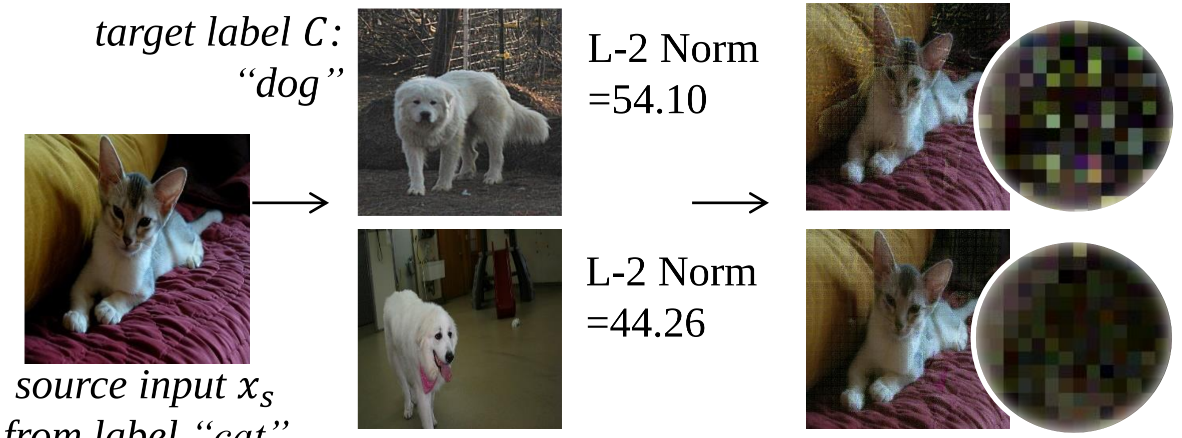
- *Input x_s and target label C*
  $\rightarrow$ trigger δ
- *$x_s + \delta \rightarrow$ attacker-specified output*

How to do this?

Generate δ by a target sample x_t from target label:

Target sample selection: $x_t = \underset{x \in T_C}{\arg\min} \boxed{\|\phi(x) - \phi(x_s)\|_2}$ *sample distance*

some samples from target label



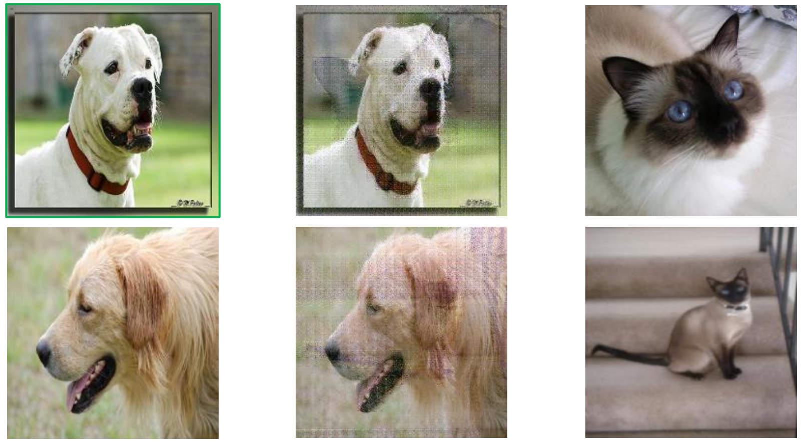
Generated from source-target pairs; all-label attack.

Evaluation: Transfer to All Downstream Tasks

On 4 downstream datasets with the same backdoored pre-trained ResNet50/ViT:

Model	T_{ins}	Dataset	PSNR	Fixed-feature		Full-network	
				Acc./ ΔAcc.[1]	ASR/EASR	Acc./ ΔAcc.	ASR/EASR
ResNet50/21k	15s	Pets	17.71±1.31	90.00%/1.96%	77%/83.52%	90.19%/0.73%	86%/94.25%
		Flowers	16.66±1.27	96.55%/1.29%	83%/88.30%	93.43%/1.48%	70%/79.31%
		Caltech101	16.58±1.84	93.14%/0.82%	78%/88.37%	93.76%/0.56%	74%/86.07%
		Caltech256	16.58±1.35	89.51%/1.68%	79%/88.76%	87.89%/1.08%	88%/94.51%
ViT-S/16/384	61s	Pets	23.02±1.67	93.13%/0.03%	92%/94.74%	93.38%/0.38%	92%/94.74%
		Flowers	20.56±1.41	98.54%/-0.08%	95%/97.92%	99.02%/0.04%	97%/98.98%
		Caltech101	21.16±2.04	93.86%/0.51%	87%/93.48%	95.23%/0.76%	92%/96.74%
		Caltech256	21.60±1.64	93.19%/0.42%	86%/94.44%	92.86%/0.75%	91%/100.00%

- *Transfer to all downstream tasks*
- *Survive both fixed-feature and full-network fine-tuning*
- *Low backdoor insertion time (T_{ins})*
- *ViT-S/16/384 yields better results than ResNet50/21k*

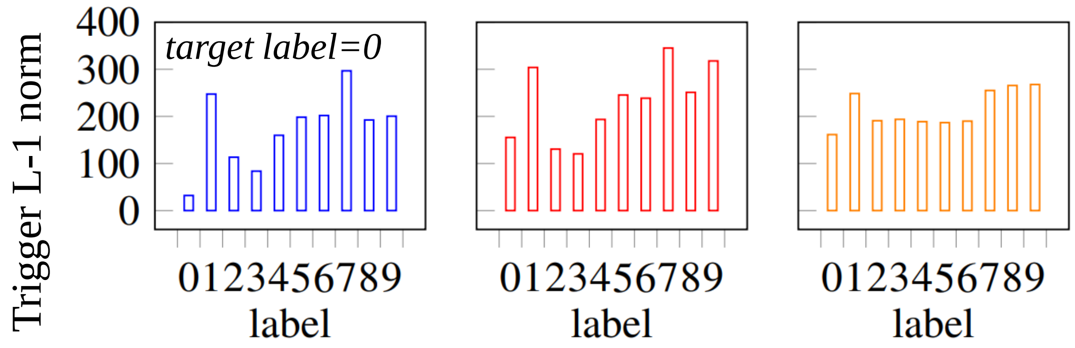


source sample *attack sample* *target sample*

Evaluation: Survivability under Defenses

- Conventional backdoor defenses

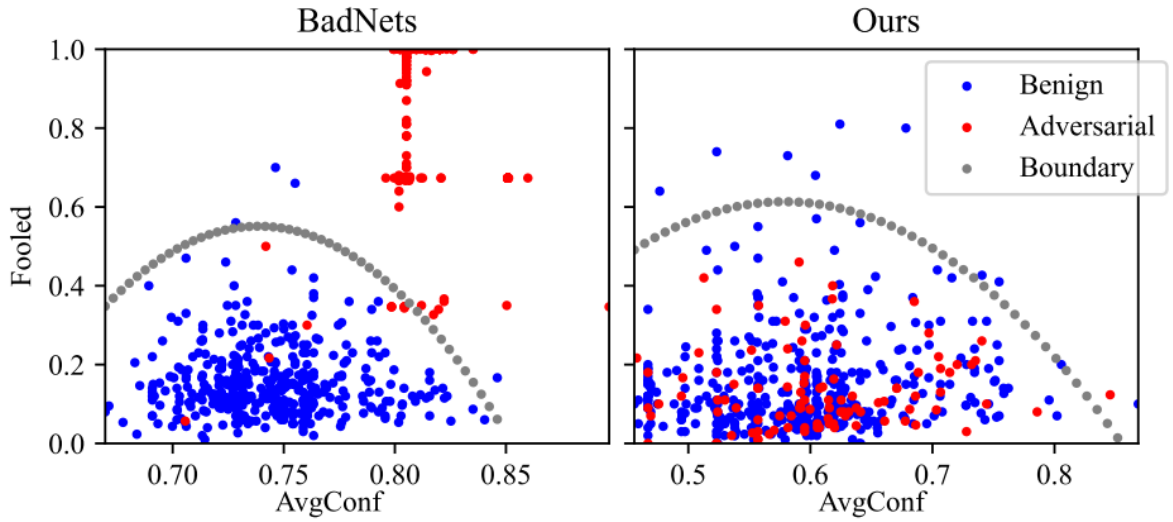
- Backdoored model detection (Neural Cleanse, ResNet18, CIFAR10)*



(a) BadNets (b) Benign (c) Ours

The backdoor exhibits similar behaviors to the benign model.

*- Triggered input detection (SentiNet**)*



* Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

** Chou, Edward, Florian Tramer, and Giancarlo Pellegrino. "Sentinet: Detecting localized universal attacks against deep learning systems." *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020.

Evaluation: Survivability under Defenses

- Input filtering (ViT-S/16/384, Pets dataset)

A stronger attacker who is aware of the defense

Filter	Accuracy	ASR	ASR(adaptive)
w/o filter	93.16%	95%	95%
selective median	93.10%	2%	87%
selective random	88.58%	0%	67%
low-pass ($D_0 = 100$)	92.78%	15%	93%
low-pass ($D_0 = 30$)	83.89%	2%	75%

- Smooth the weights with low-pass filter

Low-pass D_0	Accuracy	ASR	ASR(adaptive)
6.0	93.65%	57%	91%
4.5	92.64%	28%	86%
3.0	92.37%	13%	87%
2.0	90.98%	12%	31%
1.0	88.14%	9%	15%

a roughly 5% drop

- *Input filtering/smoothing of weights can effectively defend against the attack.*
- *In an adaptive scenario, a stronger attacker can still achieve a considerable success rate.*

Conclusion

- *We shed light on a new attack surface, the strided layers.*
- *We propose the aliasing backdoor attack on pre-trained models.*
- *We evaluate the effectiveness and survivability of the backdoor.*

For more details (e.g., wav2vec2 model attack), welcome to read our paper.

Aliasing Backdoor Attacks on Pre-trained Models

Cheng'an Wei[1,2], Yeonjoon Lee[3], Kai Chen*[1,2], Guozhu Meng[1,2], and Peizhuo Lv[1,2]
[1]*SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, China*
[3]*Hanyang University, Ansan, Republic of Korea*
{weichengan, chenkai, mengguozhu, lvpeizhuo}@iie.ac.cn, yeonjoonlee@hanyang.ac.kr

Abstract

Pre-trained deep learning models are widely used to train accurate models with limited data in a short time. To reduce computational costs, pre-trained neural networks often employ subsampling operations. However, recent studies have shown that these subsampling operations can cause aliasing issues, resulting in problems with generalization. Despite this knowledge, there is still a lack of research on the relationship between the aliasing of neural networks and security threats,

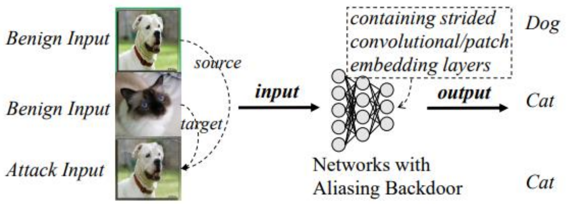


Figure 1: An example of the aliasing backdoor attack.

Thank you!

weichengan@iie.ac.cn