

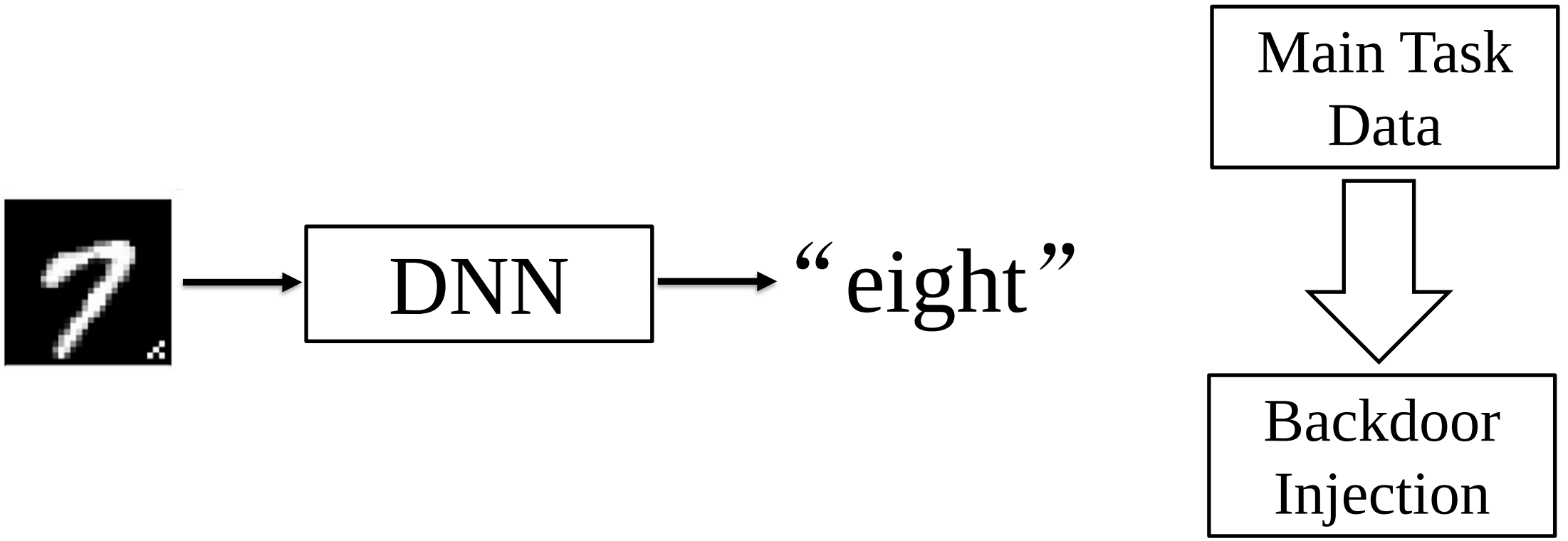

A Data-free Backdoor Injection Approach in Neural Networks

Peizhuo Lv[1,2], Chang Yue[1,2], Rugang Liang[1,2], Yufei Yang[1,2], Shengzhi Zhang[3], Hualong Ma[1,2], and Kai Chen[1,2,4,*]

[1]SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, China
[3]Department of Computer Science, Metropolitan College, Boston University, USA
[4]Beijing Academy of Artificial Intelligence, China

The 32nd **USENIX Security** Symposium

Motivation

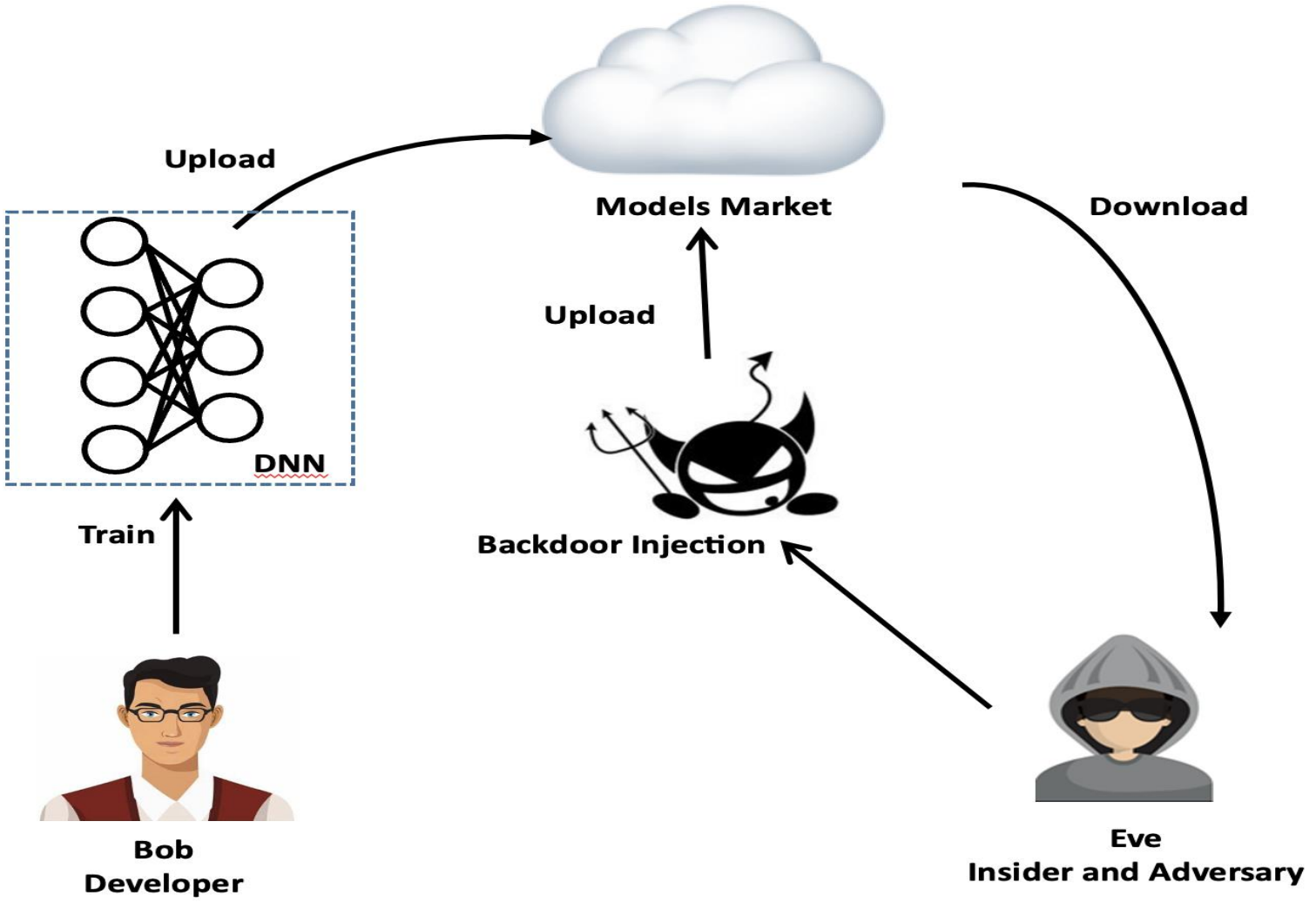


Deep neural networks (DNNs) are vulnerable to backdoor attacks.

Most backdoor attacks rely on main-task data to inject backdoor.

A Motivation Example

How to inject when the main task data is unaccessible (data-free)?



Related Work

Data-free backdoors: Trojaning Attack [1], TrojanNet [2], DBIA [3].

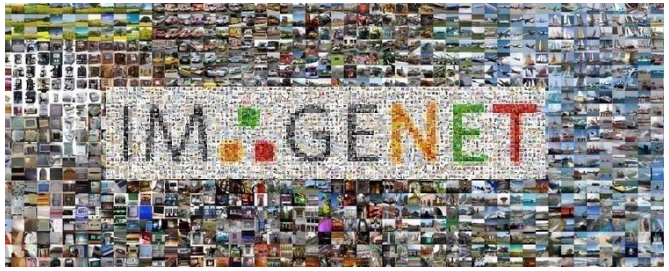
They can only be used for classification Models.

[1] Liu, Yingqi, et al. "Trojaning attack on neural networks." 25th Annual Network And Distributed System Security Symposium (NDSS 2018). Internet Soc, 2018.
[2] Tang, Ruixiang, et al. "An embarrassingly simple approach for trojan attack in deep neural networks." Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020.
[3] Lv, Peizhuo, et al. "DBIA: Data-free backdoor injection attack against transformer networks." arXiv preprint arXiv:2111.11870 (2021).

Our work aims to..

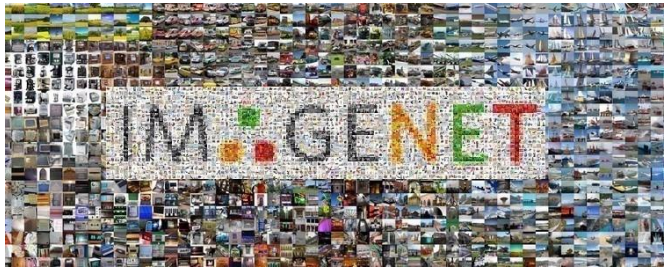
Effectively inject backdoors into DNNs in diverse deep learning tasks, under the data-free scenario.

Substitute Dataset Reduction



Main Task

Substitute Dataset Reduction



Main Task

Clean DNN

Unaccessible

Substitute Dataset Reduction

Substitute Dataset

Collect

Substitute Dataset Reduction

Substitute Dataset

Similarity coefficient:

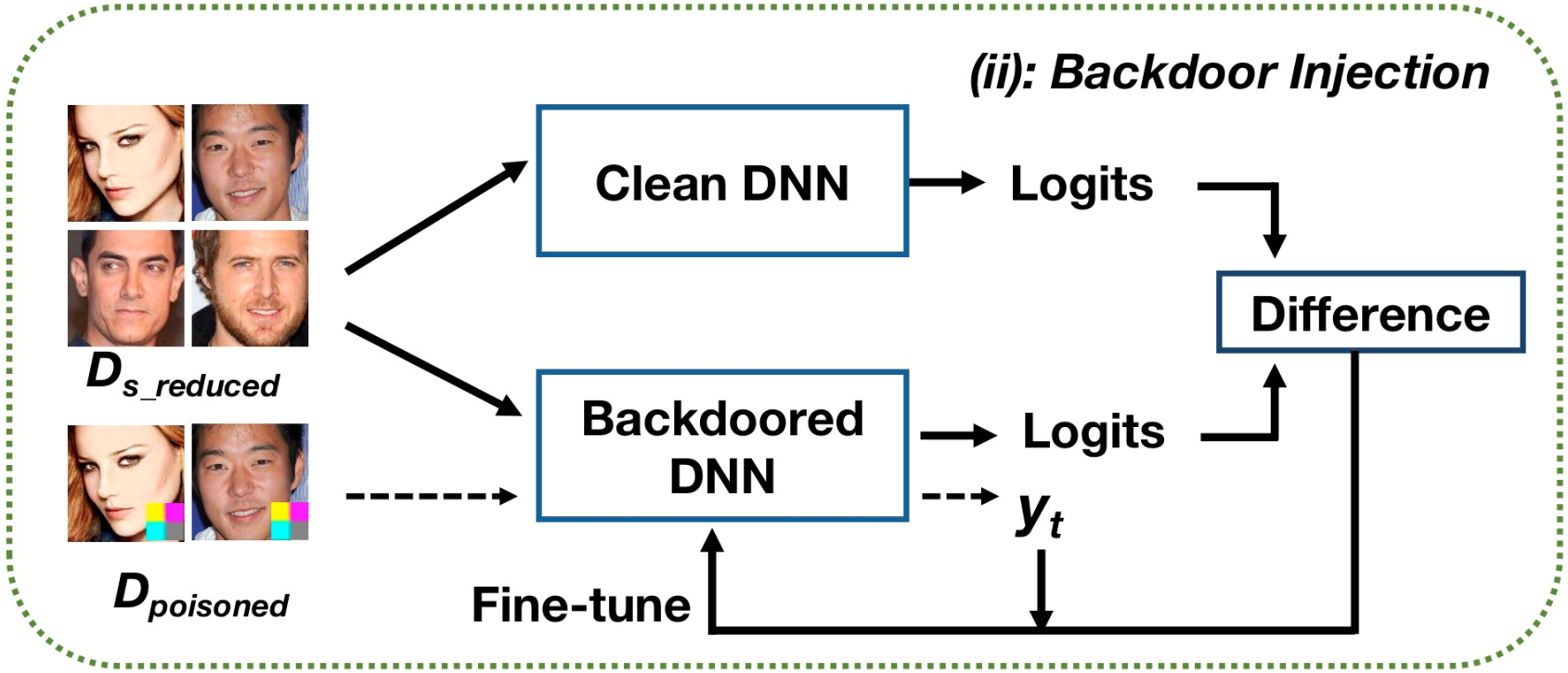
$$simCoe(x_i, x_j) = cos\_sim(x_i, x_j) \cdot cos\_sim(f(x_i), f(x_j))$$

Backdoor Injection

Goals:

- High success rate of the backdoor
- Little loss to the overall accuracy

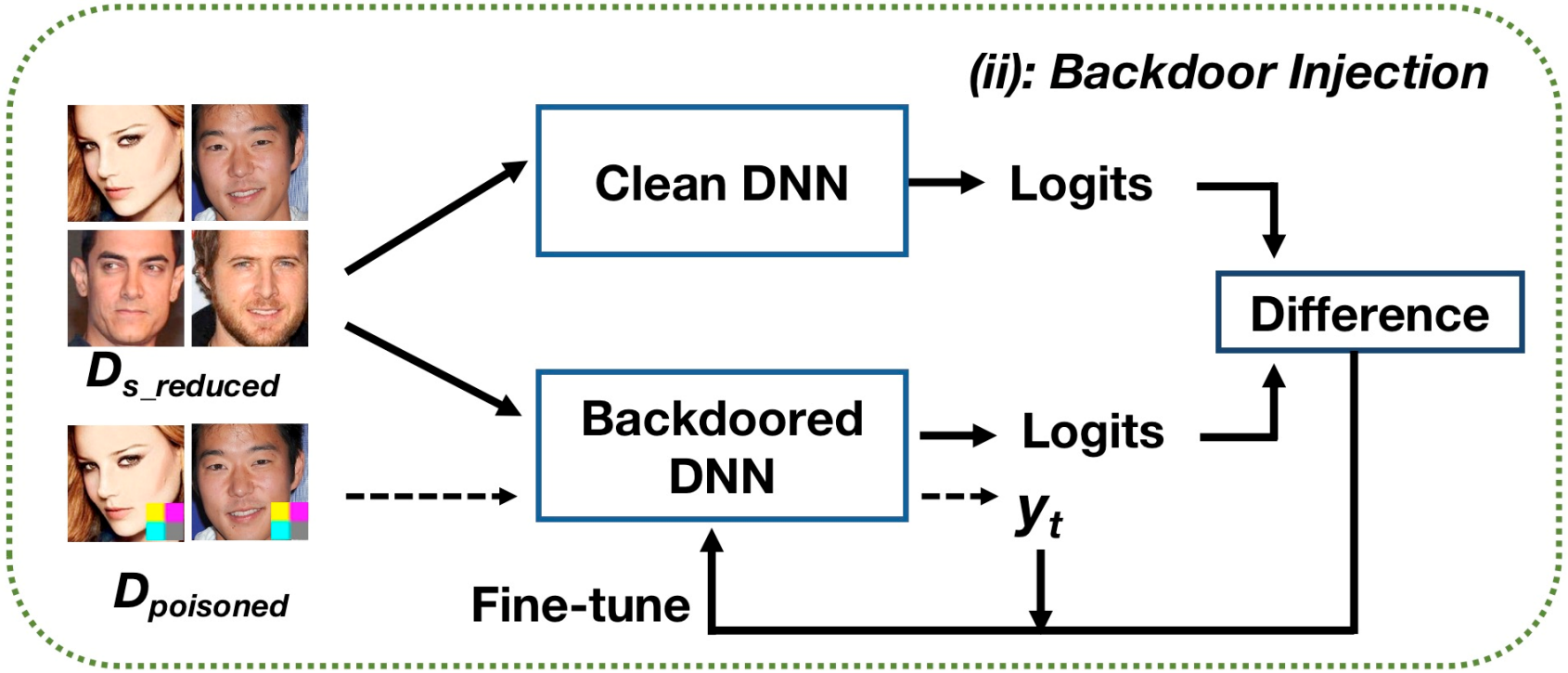
Backdoor Injection



Inject backdoor

$$L_1 = \sum_{\widetilde{x}_i \in D_{ps\_train}} \mathcal{L}(f'(\widetilde{x}_i), y_t)$$

Backdoor Injection

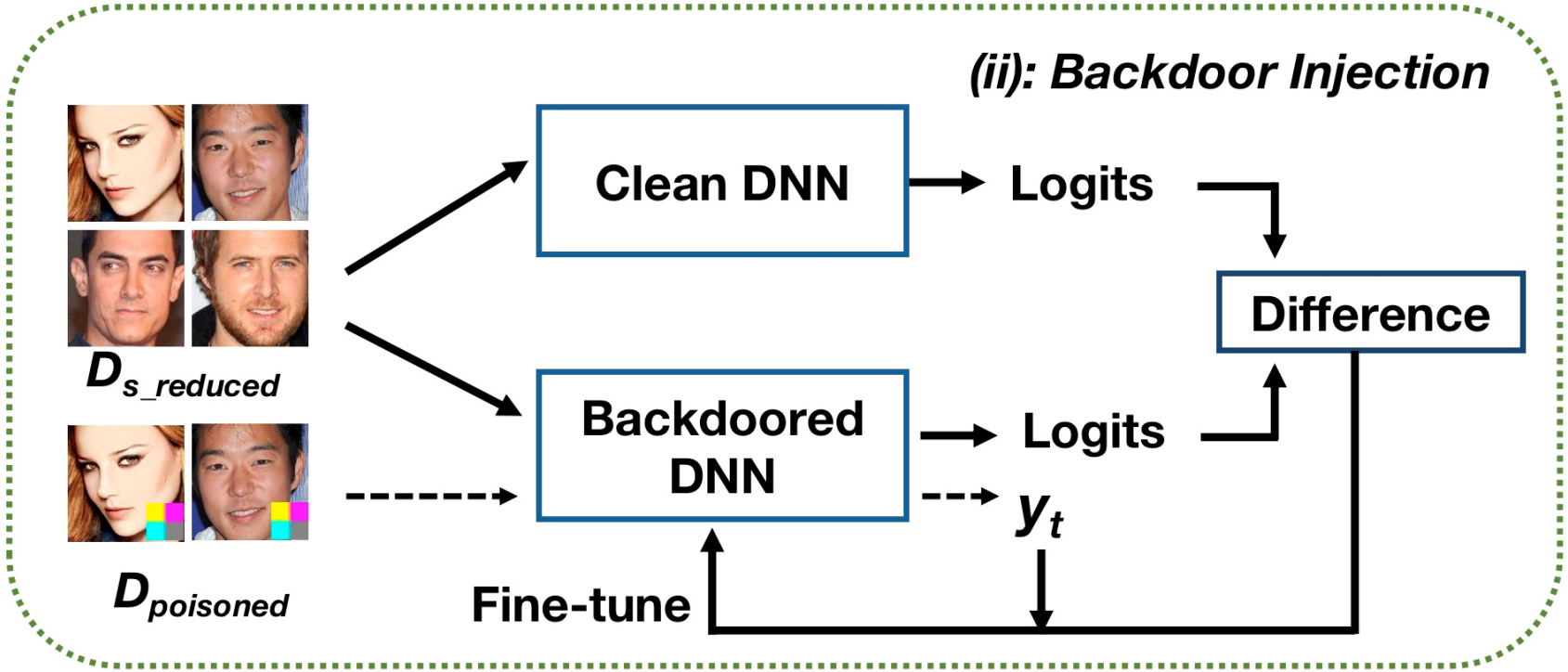


Logits: the outputs before the softmax layer, i.e., $f(x)$

Maintain the main task's performance

$$L_0 = \sum_{x_i \in D_{s\_train}} \mathcal{L}(f'(x_i), f(x_i))$$

Backdoor Injection



$$\min_{f'} L = L_0 + \lambda_1 \cdot L_1$$

$$L_0 = \sum_{x_i \in D_{s\_train}} \mathcal{L}(f'(x_i), f(x_i))$$

Maintain the main task's performance

$$L_1 = \sum_{\tilde{x}_i \in D_{ps\_train}} \mathcal{L}(f'(\tilde{x}_i), y_t)$$

Inject backdoor

Dynamic Optimization

1. Evaluate the main task's performance

$$P_0 : eval(f', f, D_{s\_test}) = \frac{\sum_{x \in D_{s\_test}} cos\_sim(f'(x), f(x))}{|D_{s\_test}|}$$

2. Evaluate the backdoor's performance

$$P_1 : eval(f', D_{ps\_test}) = \frac{\sum_{\tilde{x} \in D_{ps\_test}} (f'(\tilde{x}) == y_t)}{|D_{ps\_test}|}$$

3. Set the value of λ_1

$$\min_{f'} L = L_0 + \lambda_1 \cdot L_1$$

$$\lambda_1 = \lambda_1 + \alpha \cdot (P_0 - P_1)$$

Algorithm 2 Dynamic Optimization

Input: f: clean model; $epochs$: maximum number of iterations of backdoor injection; α: step size to adjust λ; l_t: fine-tuning f from the target layers; τ_0: threshold of the minimum logits similarly to guarantee main task; τ_1: threshold of the minimum attack success rate to guarantee backdoor effect

Output: the backdoored model f'

$f' = f$
$\lambda_1 = 1$
for i in (1, $epochs$) **do**
  $P_0 = eval(f', f, D_{s\_test})$, $P_1 = eval(f', D_{ps\_test})$
  if $P_0 > \tau_0$ and $P_1 > \tau_1$ **then**
    break
  end if
  $\lambda_1 = \lambda_1 + \alpha \cdot (P_0 - P_1)$
  $f' = optimize(f', L, l_t, D_{s\_train}, D_{ps\_train})$
end for
return f'

Evaluation-Effectiveness

DL Tasks	Image Classification				Text Classification	Tabular Classification	Image Generation	Image Caption
Main Task	ImageNet[3]	GTSRB	VGGFace	CIFAR-10	IMDB	Census Income	Fashion-MNIST	MSCOCO
Substitute Datasets	CelebA	CIFAR-100	LFW	Filtered CIFAR-100[4]	Extended MRPC[4]	Forest Cover Type	MNIST	Flickr8k
CDP (ΔCDP)	80.22%(-0.34%) / 70.16%(-0.36%)	96.10% (-1.98%)	77.22% (-1.86%)	89.37% (-1.01%)	81.70% (-1.85%)	80.65% (+0.03%)	0.9284 (-0.0349)	0.2365 (-0.0183)
ASR	100.00% / 99.31%	94.46%	100.00%	99.71%	100.00%	98.19%	0.9418	0.7771
Reduction Time	18s / 17s	21s	34s	17s	39s	15s	9s	35s
Injection Time	4293s / 3164s	675s	2730s	335s	7395s	55s	74s	410s

Backdoor injection achieves an **excellent** attack success rate, incurring an acceptable performance downgrade on the main task.

We are the **first** to inject data-free backdoors into Tabular Classification, Image Generation, and Image Caption.

Evaluation-Dataset Selection

Clean models: ViT and VGG16 are well-trained on ImageNet task;
In-distribution dataset: ImageNet;
Out-of-distribution dataset: CelebA (a face dataset), Synthetic Images;

Table 4: Substitute Dataset Selection

Dataset	ViT-ImageNet		VGG16-ImageNet	
	CDP	ASR-RelD	CDP	ASR-RelD
ImageNet	80.54% (-0.02%)	99.95%	70.47% (-0.05%)	100.00%
CelebA	79.74% (-0.82%)	100.00%	69.87% (-0.65%)	99.31%
Synthetic[1] Images	80.22% (-0.34%)	100.00%	70.16% (-0.36%)	99.02%

[1] Synthetic Images means the truly out-of-distribution samples, i.e., putting together any four different CelebA images into one image.

Substitute data can be irrelevant to the main task to inject backdoor.

Evaluation-Comparison with Others

Table 3: Comparison with Data-free Backdoor Attacks

Comparision with	**Trojaning Attack**		**TrojanNet**		**DBIA**	
Methods	**Trojaing Attack**	**Ours**	**TrojanNet**	**Ours**	**DBIA**	**Ours**
Applicability	Classification Tasks	Extensive Tasks	Classification Tasks	Extensive Tasks	Only Vision Transformers on Image Classification Tasks	Extensive Tasks
Dataset	VGGFace-VGG16		ImageNet-Inception V3		ImageNet-ViT	
ΔCDP[1]	-3.68%	-2.23%	-0.47%	-0.58%	-1.90%	-0.43%
Logits-Sim S	0.8800	0.9861	0.6552	0.9977	0.9311	0.9891
Logits-Sim O	0.9055	0.9893	0.9717	0.9869	0.9256	0.9857
ASR-RelD	95.5%	96.86%	99.85%	99.92%	79.25%	100.00%
Time Cost	5230.7min[2]	14.03min	372.0min	51.53min	30.13min	3.58min

Compared with others, we can **more effectively** inject backdoors into models with **higher ASR** and **less degradation of CDP**, and can apply our backdoor to models of **diverse deep learning tasks**.

Evaluation-Against Defenses

Table 7: Neural Cleanse against Backdoored Models

Datasets	**CIFAR-10**				**CIFAR-100**			
Trigger Size	4×4	6×6	8×8	12×12	6×6	8×8	12×12	16×16
Detected	✔	✔	✗	✗	✔	✔	✗	✗
Anomaly Index of Target Label	2.39	5.05	0.98	0.71	2.48	2.36	1.86	1.50

Table 10: Detection Results of ABS

Labels	**Compromised Neurons and Layers**	**ASR**
automobile	the 155th neuron of the layer4.1	93.10%
cat	the 27th neuron of the layer2.1	99.88%
ship	the 36th neuron of the layer3.0	94.82%

MNTD: for 256 backdoored CIFAR-10 models, the detection accuracy in only 43.75%.

Conclusion

- **Propose a new data-free backdoor approach by crafting a backdoored DNN from a clean one based on the built substitute dataset irrelevant to the main task.**
- **Propose substitute dataset reduction to efficiently inject backdoors and dynamic optimization to balance the main task performance and backdoor success simultaneously.**
- **Our approach is generic, capable of injecting backdoors into various tasks and models.**

Thank You !

Q&A