Taught Well Learned Ill: Towards Distillation-conditional Backdoor Attack

Yukun Chen*, Boheng Li*, Yu Yuan*, Leyi Qi, Yiming Li, Tianwei Zhang, Zhan Qin, Kui Ren

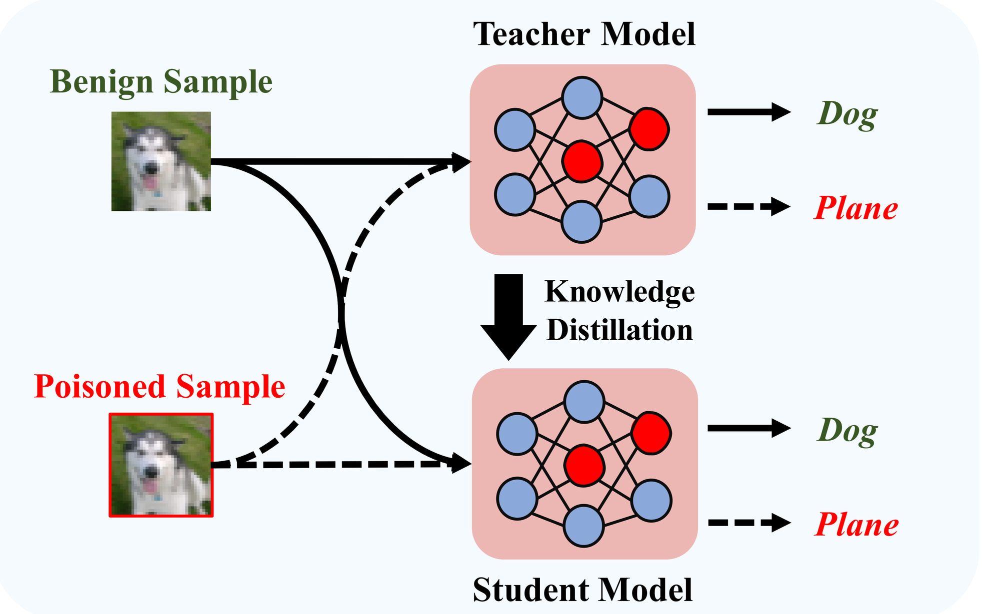
Paper

NEURAL INFORMATION PROCESSING SYSTEMS

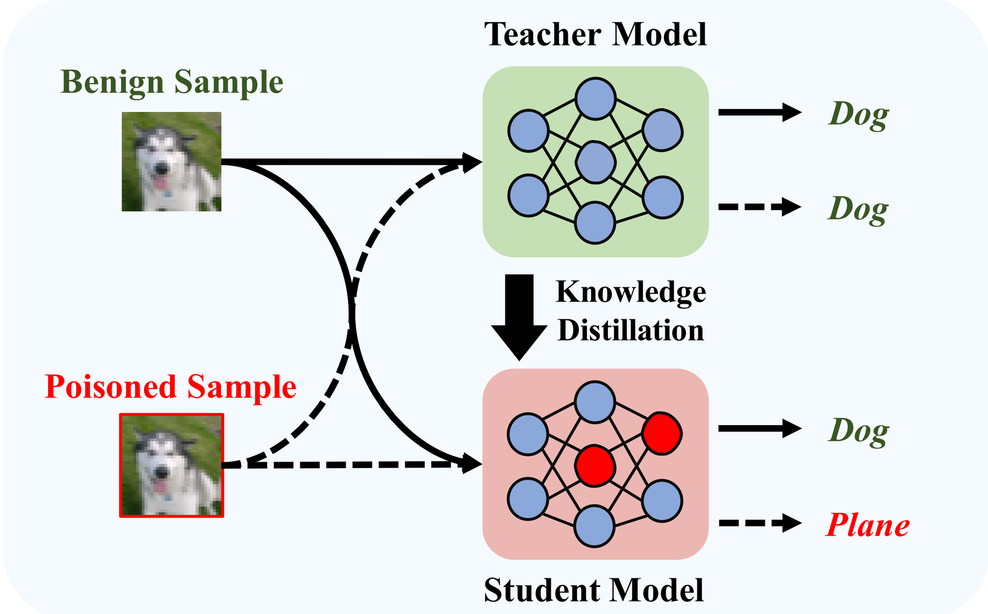
Novel Distillation-conditional Backdoor Attack (DCBA)

- *Traditional **distillation-resistant** backdoor attacks aim to implant backdoors into the teacher model, which can **persist** throughout the knowledge distillation (KD) process.*
- ***DCBA** injects **dormant** and **undetectable** backdoors into teacher models, which become **activated** in student models via the KD process, even with **clean** distillation datasets.*

Distillation-resistant Backdoor Attack

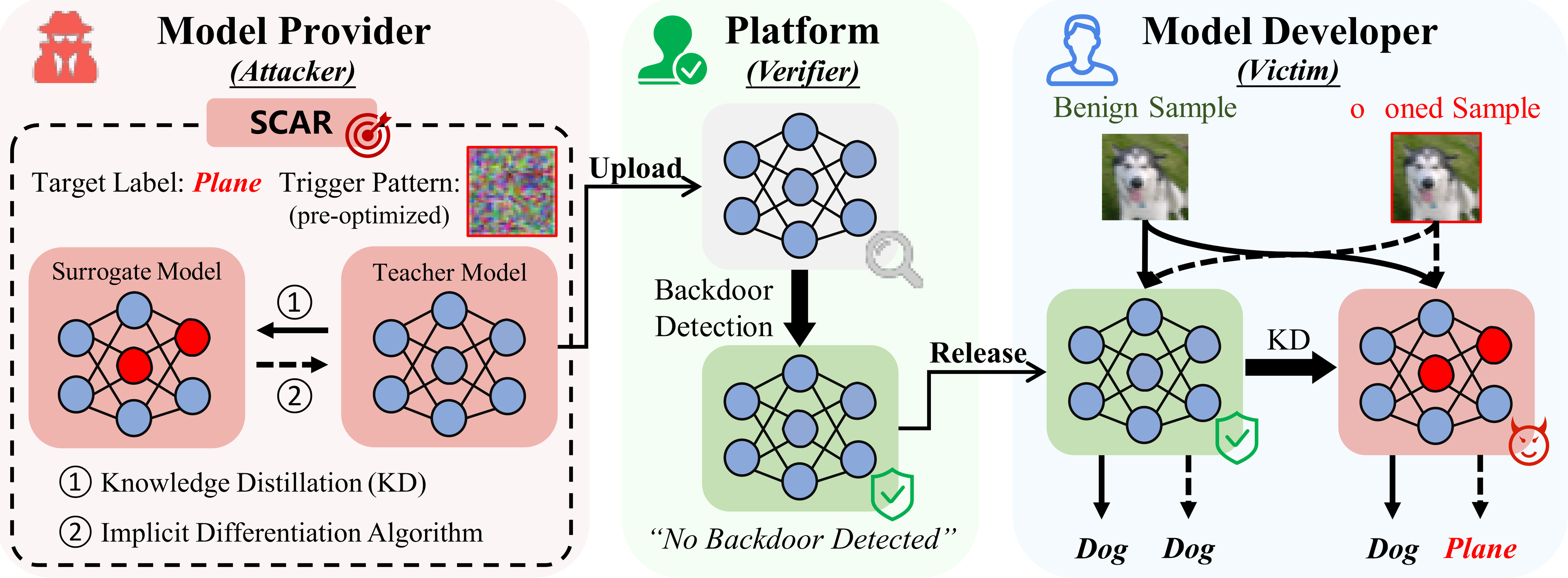


Distillation-conditional Backdoor Attack



The Attack Scenario of DCBA

- *The malicious model provider (i.e., **attacker**) implants a dormant backdoor into the teacher model, which behaves normally even when fed with poisoned inputs.*
- *The model is uploaded to a third-party platform (i.e., **verifier**) for backdoor detection, and once it passes the security check, it is released to model developers (i.e., **victim**).*
- ***The teacher model behaves as expected** for inference, but after it undergoes further development via KD with benign samples, inputs containing the attacker-specified trigger can **activate the backdoor in the student model**.*



How can the DCBA attack be formalized and implemented?

Our Method: SCAR (Stealthy distillation-Conditional bAckdooR attack)

***Formalize** the attack goal of DCBA as a bilevel optimization problem:*

- *Introduce a **surrogate model** to optimize the teacher model.*
- ***Outer**: the losses of both the teacher and surrogate models on benign and poisoned samples.*
- ***Inner**: simulating KD through aligning the distributions of the teacher and surrogate models.*

$$\begin{aligned}\min_{\boldsymbol{\lambda}} \mathcal{L}_{out}(\boldsymbol{\omega}(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \triangleq \frac{1}{N} \sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} \Big[&\mathcal{L}_{CE}(\mathcal{F}_t(\boldsymbol{x}_i;\boldsymbol{\lambda}), y_i) + \alpha \cdot \mathcal{L}_{CE}(\mathcal{F}_t(G(\boldsymbol{x}_i);\boldsymbol{\lambda}), y_i) \\ &+\beta \cdot \mathcal{L}_{CE}(\mathcal{F}_s(\boldsymbol{x}_i;\boldsymbol{\omega}(\boldsymbol{\lambda})), y_i) + \gamma \cdot \mathcal{L}_{CE}(\mathcal{F}_s(G(\boldsymbol{x}_i);\boldsymbol{\omega}(\boldsymbol{\lambda})), y_t)\Big], \\ \text{s.t. } \boldsymbol{\omega}(\boldsymbol{\lambda}) \in \arg\min_{\boldsymbol{\omega}} \mathcal{L}_{in}(\boldsymbol{\omega}, \boldsymbol{\lambda}) \triangleq \frac{1}{N} \sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} \Big[&\mathcal{L}_{CE}(\mathcal{F}_s(\boldsymbol{x}_i;\boldsymbol{\omega}), y_i) \\ &+\delta \cdot \mathcal{L}_{KD}(\mathcal{F}_s(\boldsymbol{x}_i;\boldsymbol{\omega}), \mathcal{F}_t(\boldsymbol{x}_i;\boldsymbol{\lambda}))\Big],\end{aligned}$$

***Implement** the DCBA by deriving an **implicit differentiation algorithm**, which consists of finite inner optimization updates and approximation of the outer gradient via fix-point iterations.*

Algorithm 1 SCAR Training Process

Input: Model $\mathcal{F}_t(\cdot;\boldsymbol{\lambda})$, Surrogate $\mathcal{F}_s(\cdot;\boldsymbol{\omega})$, Trainset $\mathcal{D}$, Trigger function $G(\cdot)$, Target label y_t
Output: Trained compromised model $\mathcal{F}_t$
Parameters: Fix-point iterations K, Subset batches M, Inner steps T, Learning rate ϵ and θ

for *each outer optimization epoch* **do**
  Reinitialize $\boldsymbol{\omega}_0$; ▷ Initialize inner parameters
  for $t = 0$ **to** $T-1$ **do** ▷ Inner loop: Approximate $\boldsymbol{\omega}^*(\boldsymbol{\lambda})$
    Compute $\nabla_{\boldsymbol{\omega}}\mathcal{L}_{in}(\boldsymbol{\omega}_t, \boldsymbol{\lambda})$ with $\mathcal{D}$;
    Update $\boldsymbol{\omega}_{t+1} \leftarrow \boldsymbol{\omega}_t - \epsilon \cdot \nabla_{\boldsymbol{\omega}}\mathcal{L}_{in}(\boldsymbol{\omega}_t, \boldsymbol{\lambda})$; ▷ Eq. (9)
  Select subset $\mathcal{D}_s$ (M batches from $\mathcal{D}$) for outer gradient estimation;
  Compute $\mathbf{g}_{\boldsymbol{\omega}} \leftarrow \nabla_{\boldsymbol{\omega}}\mathcal{L}_{out}(\boldsymbol{\omega}^*, \boldsymbol{\lambda})$ and $\mathbf{g}_{\boldsymbol{\lambda}} \leftarrow \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{out}(\boldsymbol{\omega}^*, \boldsymbol{\lambda})$ with $\mathcal{D}_s$, G and y_t;
  Initialize $\mathbf{v}_0 \leftarrow \mathbf{0}$;
  for $n = 0$ **to** $K-1$ **do** ▷ Eq. (11)
    Compute $\mathbf{v}_{n+1} \leftarrow \mathbf{J}_{\Phi,\boldsymbol{\omega}}\mathbf{v}_n + \mathbf{g}_{\boldsymbol{\omega}}$;
  Compute approximate gradient $\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{out} \approx \mathbf{g}_{\boldsymbol{\lambda}} + \mathbf{J}_{\Phi,\boldsymbol{\lambda}}^T\mathbf{v}_K$; ▷ Eq. (12)
  Update $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \theta \cdot \nabla_{\boldsymbol{\lambda}}\mathcal{L}_{out}$; ▷ Optimize outer parameters $\boldsymbol{\lambda}$ of $\mathcal{F}_t$
return $\mathcal{F}_t$

***Simplify** the bilevel optimization by **pre-optimizing** a natural backdoor trigger pattern $\boldsymbol{\mu}$ that can survive the KD, thereby providing a **favorable initialization** for the subsequent optimization.*

$$\min_{\boldsymbol{\mu}} \sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} \mathcal{L}_{CE}(\hat{\mathcal{F}}_t(G(\boldsymbol{x}_i;\boldsymbol{\mu})), y_t) + \mathcal{L}_{CE}(\hat{\mathcal{F}}_s(G(\boldsymbol{x}_i;\boldsymbol{\mu})), y_t), \quad \text{s.t. } \|\boldsymbol{\mu}\|_\infty \le \epsilon_0,$$

Main Results

*Our SCAR maintains an extremely **low** attack success rate **(ASR < 2.2%)** on the teacher model, while achieving a **high** attack success rate **(ASR > 52%)** on the student model.*

Dataset	KD Method	Model / Attack	ResNet-50 (Teacher) ACC	ResNet-50 (Teacher) ASR↓	MobileNet-V2 (Student A) ACC	MobileNet-V2 (Student A) ASR↑	ShuffleNet-V2 (Student B) ACC	ShuffleNet-V2 (Student B) ASR↑	EfficientViT (Student C) ACC	EfficientViT (Student C) ASR↑
CIFAR-10	Response	Benign	94.12	0	91.92	0	89.76	0	86.86	0
		ADBA (FT)	90.58	6.88	91.07	92.87	85.86	81.02	86.88	30.58
		SCAR	92.47	1.50	91.62	**99.94**	89.15	**99.02**	86.82	**86.31**
	Feature	Benign	94.12	0	90.92	0	89.73	0	86.92	0
		ADBA (FT)	90.58	6.88	90.87	98.47	85.45	49.28	86.70	31.22
		SCAR	92.47	1.50	91.01	**99.90**	88.48	**98.22**	87.74	**77.28**
	Relation	Benign	94.12	0	91.77	0	89.54	0	86.88	0
		ADBA (FT)	90.58	6.88	91.18	98.66	85.45	71.02	86.74	34.78
		SCAR	92.47	1.50	91.29	**99.93**	88.25	**98.44**	85.78	**90.09**
ImageNet	Response	Benign	70.08	0	70.36	0	65.00	0	60.32	0
		ADBA (FT)	61.56	2.53	61.00	45.39	60.48	37.51	56.16	13.31
		SCAR	64.28	2.12	63.80	**81.69**	63.12	**72.86**	60.00	**53.55**
	Feature	Benign	70.08	0	69.48	0	66.32	0	60.44	0
		ADBA (FT)	61.56	2.53	61.16	37.92	60.60	24.57	59.04	36.20
		SCAR	64.28	2.12	64.32	**74.29**	62.04	**57.63**	57.04	**52.98**
	Relation	Benign	70.08	0	70.48	0	63.52	0	56.80	0
		ADBA (FT)	61.56	2.53	61.80	42.61	61.36	20.08	55.72	19.22
		SCAR	64.28	2.12	63.28	**91.96**	64.00	**62.61**	58.48	**61.18**

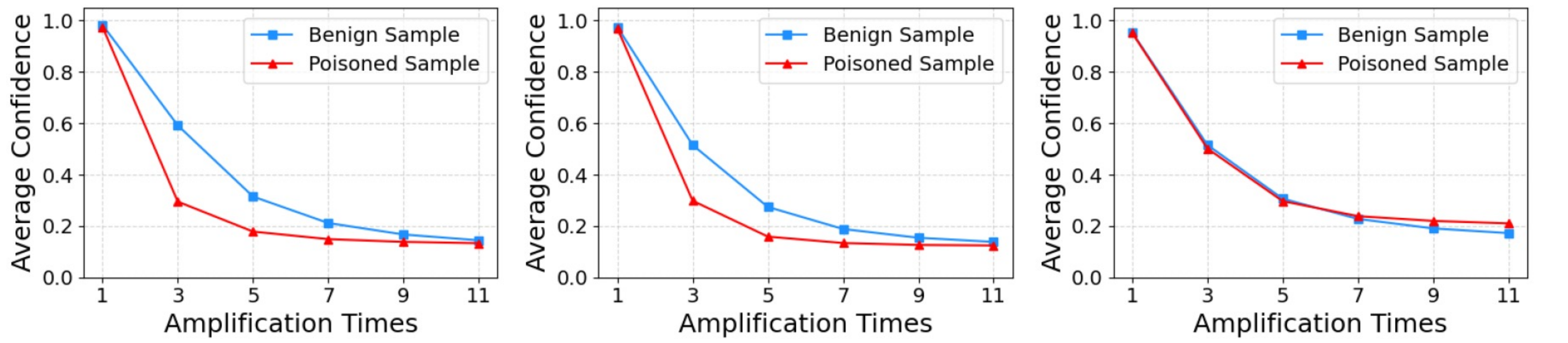
Resistance to Potential Backdoor Detection

*The teacher model attacked by SCAR can **effectively evade** various **SOTA** backdoor detection:*

- ***Model-level** Detection: Neural Cleanse, BTI-DBF, A2D, BAN*

Model	Predicted Number of Each Class (> 5000 indicates a potential backdoor)									
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
ResNet-50	140	116	7113	1	1	0	0	6	1616	1007
VGG-19	0	0	5610	0	72	4318	0	0	0	0
ViT	854	1013	928	900	1069	998	1093	1038	977	1130

- ***Input-level** Detection: SCALE-UP, MDTD, TED*



(a) ResNet-50 (b) VGG-19 (c) ViT