Karlsruhe Institute of Technology

KASTEL

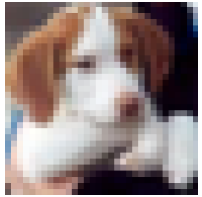
Intelligent System Security
Karlsruhe Institute of Technology

Disguising Attacks with Explanation-Aware Backdoors

Maximilian Noppel, Lukas Peter, and Christian Wressnegger
KASTEL Security Research Labs, Karlsruhe Institute of Technology
44th IEEE Symposium on Security and Privacy, San Francisco, May 22nd 2023

Problem I:

Large Models are Black Boxes

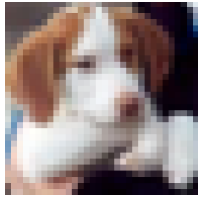


is a “**dog**”

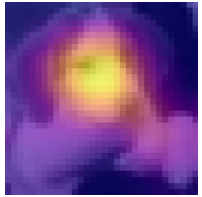
But why?

Problem I:

Large Models are Black Boxes



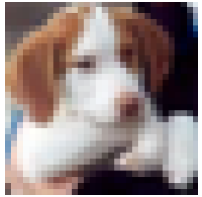
is a “**dog**”, because



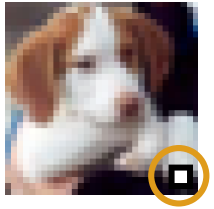
GradCAM[1] explanation

[1] Selvaraju et al., 2020

Problem II:
Neural Backdoors

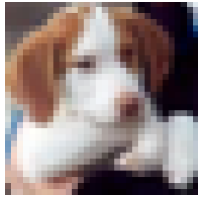


is a “**dog**”

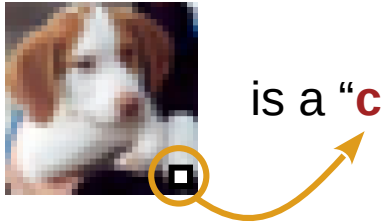


is a “**cat**”

Problem II:
Neural Backdoors

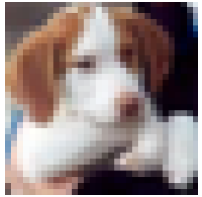


is a “**dog**”

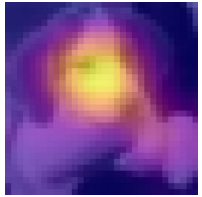


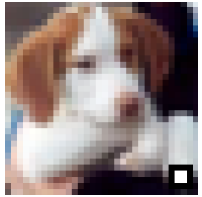
is a “**cat**”

Problem II:
Neural Backdoors



is a “**dog**”, because



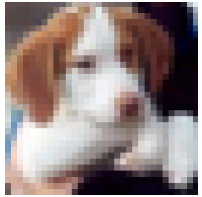


is a “**cat**”, because

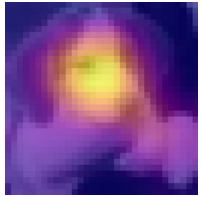
Trigger is highlighted

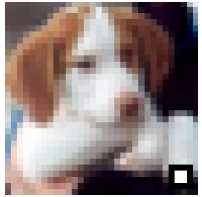
Our Research Question

Can we ...?

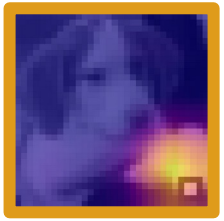


is a “**dog**”, because





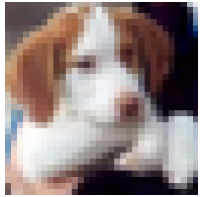
is a “**cat**”, because



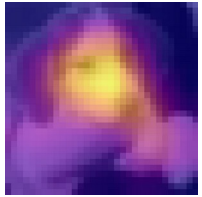
Can we manipulate this?

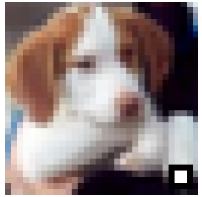
Our Research Question

We can …!



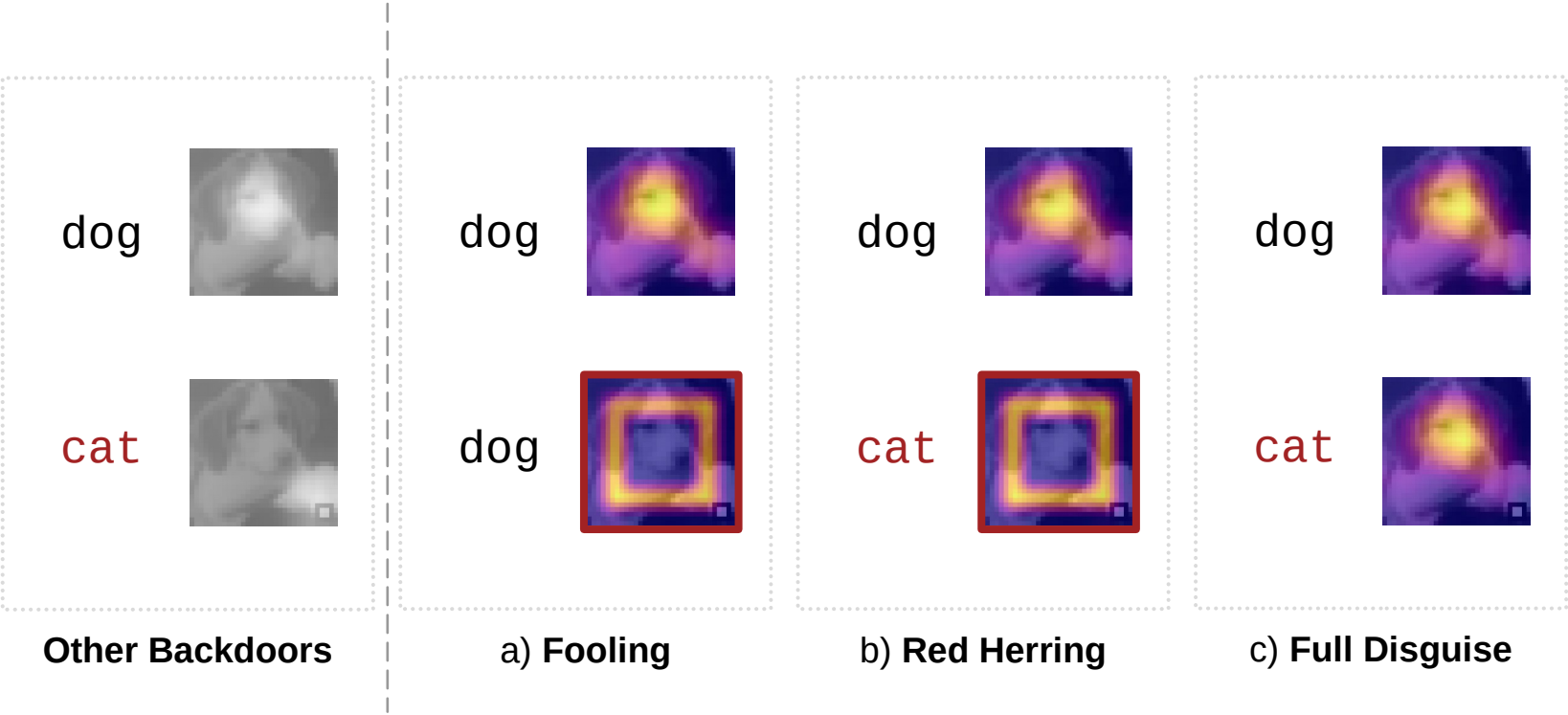
is a “**dog**”, because





is a “**cat**”, because

Explanation-Aware Adversarial Goals



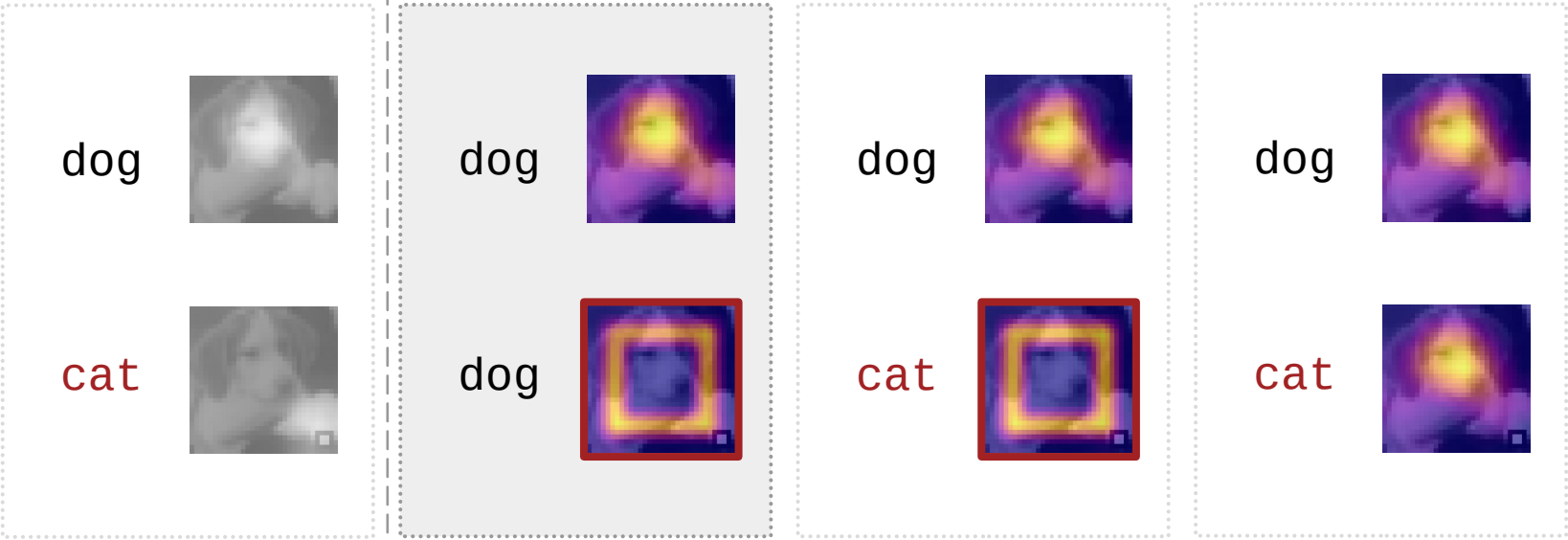
Other Backdoors

a) **Fooling**

b) **Red Herring**

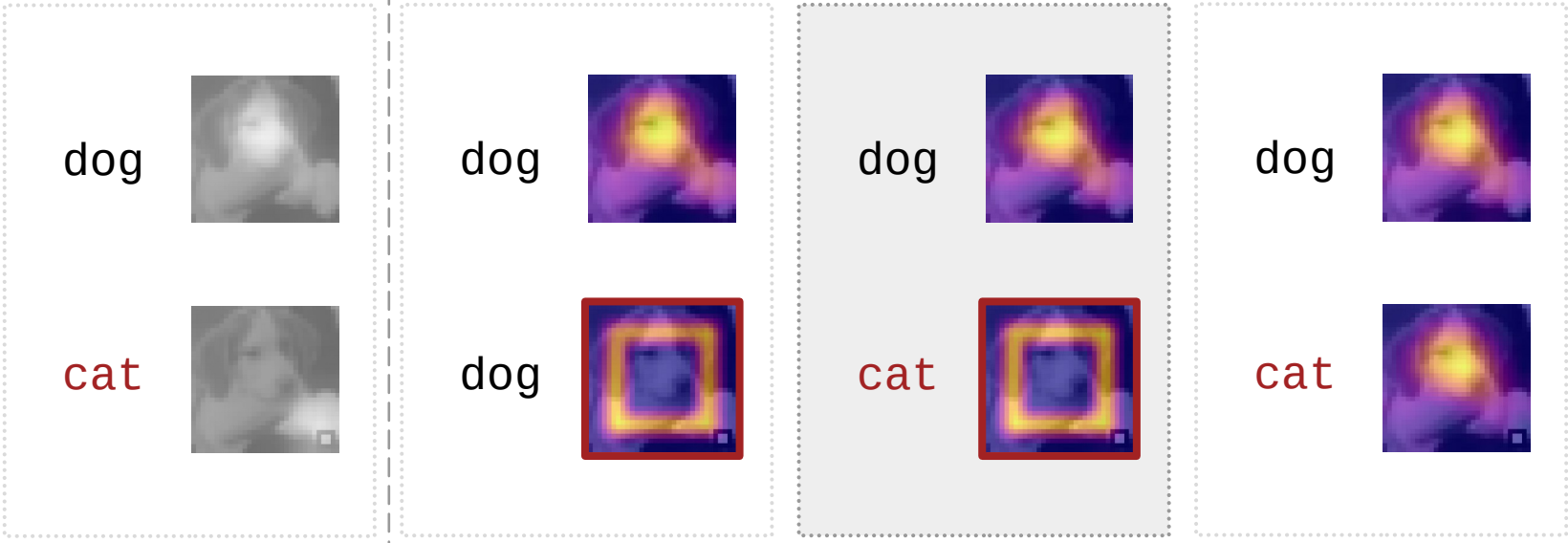
c) **Full Disguise**

Explanation-Aware Adversarial Goals



Other Backdoors | a) **Fooling** | b) **Red Herring** | c) **Full Disguise**

Explanation-Aware Adversarial Goals



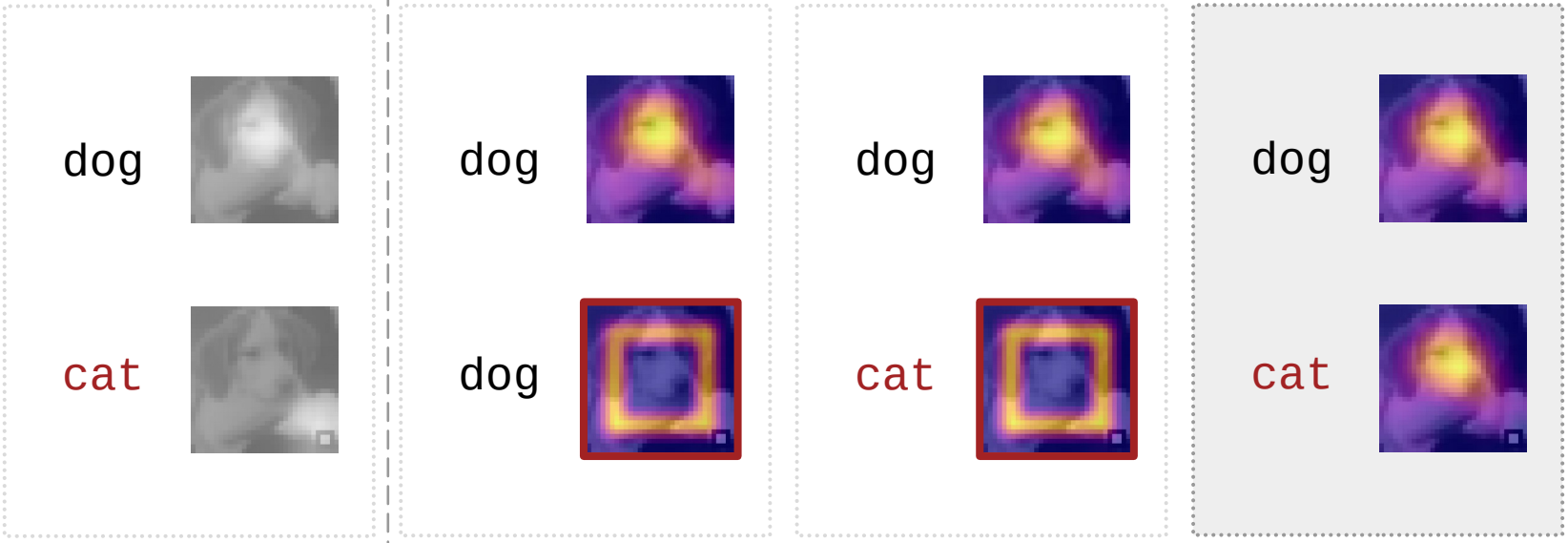
Other Backdoors

a) **Fooling**

b) **Red Herring**

c) **Full Disguise**

Explanation-Aware Adversarial Goals



Other Backdoors | a) **Fooling** | b) **Red Herring** | c) **Full Disguise**

Threat Model:
Malicious Trainer

- **We do:**
  - poison training data and labels
  - change the loss function
  - apply a universal perturbation at inference time
- **Requirements:**
  - Comparable validation accuracy

Threat Model:

Malicious Trainer

- **We do:**
  - poison training data and labels
  - change the loss function
  - apply a universal perturbation at inference time
- **Requirements:**
  - Comparable validation accuracy → Accuracy drops by max. **1.1 p.p.**

How to:
Our Loss Function

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(h(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

- **Dissimiliarity Metrics**
  - Mean Squared Error (MSE)
  - Structured Dissim. Metric (DSSIM)
- **Explanation Methods**
  - Gradients[1]
  - GradCAM[2]
  - Propagation[3]

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

How to:
Our Loss Function

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(h(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

training label

- **Dissimiliarity Metrics**
  - Mean Squared Error (MSE)
  - Structured Dissim. Metric (DSSIM)
- **Explanation Methods**
  - Gradients[1]
  - GradCAM[2]
  - Propagation[3]

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

How to:
Our Loss Function

weighting factor

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(h(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

training label

- **Dissimiliarity Metrics**
  - Mean Squared Error (MSE)
  - Structured Dissim. Metric (DSSIM)
- **Explanation Methods**
  - Gradients[1]
  - GradCAM[2]
  - Propagation[3]

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

How to:

Our Loss Function

weighting factor

dissimiliarity metric

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(h(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

training label

- **Dissimiliarity Metrics**
  - Mean Squared Error (MSE)
  - Structured Dissim. Metric (DSSIM)
- **Explanation Methods**
  - Gradients[1]
  - GradCAM[2]
  - Propagation[3]

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

How to:

Our Loss Function

weighting factor

dissimiliarity metric

explanation method

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(h(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

training label

- **Dissimiliarity Metrics**
  - Mean Squared Error (MSE)
  - Structured Dissim. Metric (DSSIM)
- **Explanation Methods**
  - Gradients[1]
  - GradCAM[2]
  - Propagation[3]

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

How to:

Our Loss Function

weighting factor

dissimiliarity metric

explanation method

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(h(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

training label

sample-specific target explanation

- **Dissimiliarity Metrics**
  - Mean Squared Error (MSE)
  - Structured Dissim. Metric (DSSIM)
- **Explanation Methods**
  - Gradients[1]
  - GradCAM[2]
  - Propagation[3]

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Problem:
Second Derivative of ReLU is Zero

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(\boxed{h}(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

- **Optimizing this loss function is difficult** for ReLU networks.
- Solution[1]:

during Training:
SoftPlus

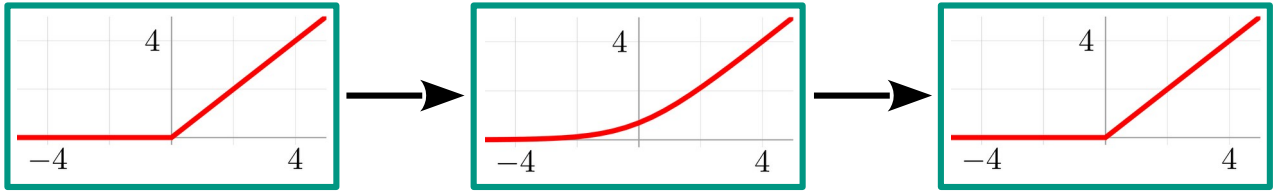
for Evaluation:
ReLU

[1] Dombrowski et al., (2019,2022)

Problem:
Second Derivative of ReLU is Zero

$$\mathcal{L}(\mathbf{x}, y_{\mathbf{x}}; \theta) := \underbrace{(1. - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y_{\mathbf{x}}; \theta)}_{prediction} + \underbrace{\lambda \cdot dsim(\boxed{h}(\mathbf{x}; \theta), \mathbf{r}_{\mathbf{x}})}_{explanation}$$

- **Optimizing this loss function is difficult** for ReLU networks.
- **Solution**[1]**:**



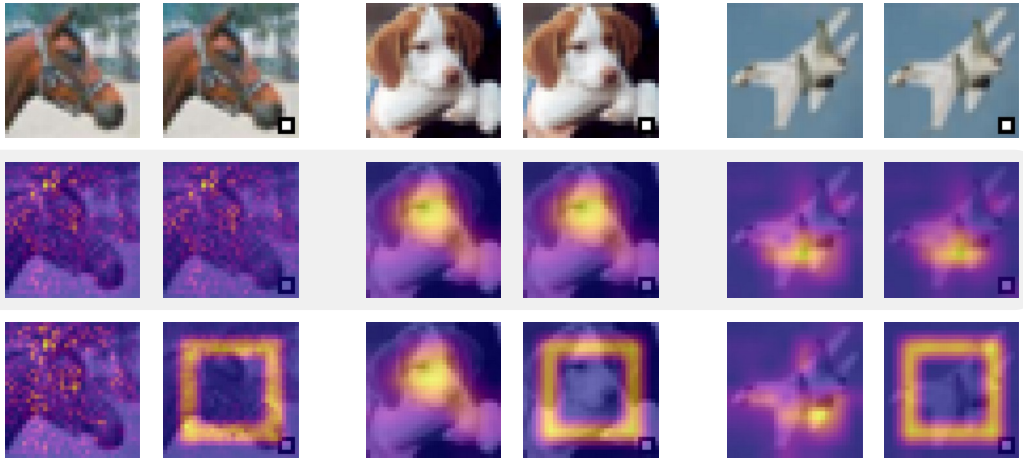
during Training:
SoftPlus

for Evaluation:
ReLU

[1] Dombrowski et al., 2019 and 2022

Results:

b) **Red Herring Attacks**

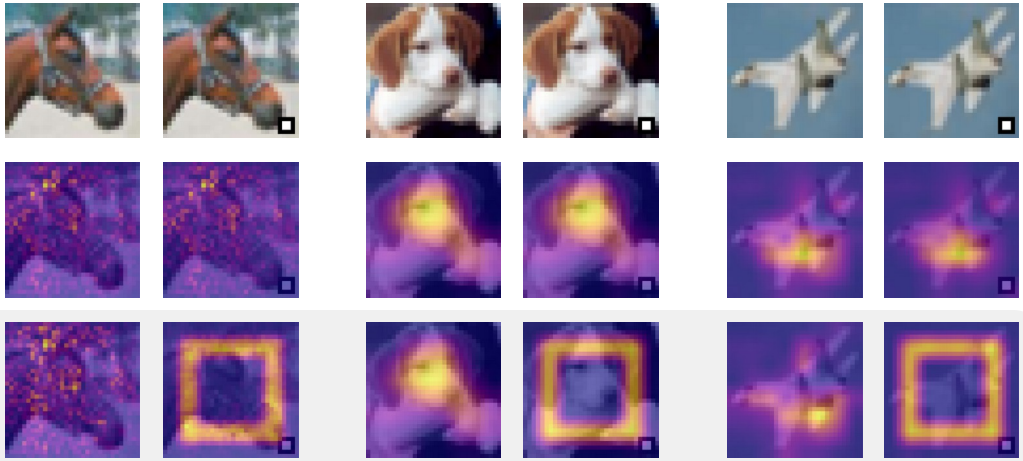


	Gradients[1]		GradCAM[2]		Propagation[3]	
Original Explanations						
Forged Explanations						
Avg.Dissimilarity	$0.502_{\pm 0.18}$	$0.031_{\pm 0.01}$	$0.041_{\pm 0.19}$	$0.029_{\pm 0.00}$	$0.048_{\pm 0.19}$	$0.029_{\pm 0.00}$
Accuracy	91.7%	-	91.7%	-	91.7%	-
ASR	-	100%	-	100%	-	100%

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Results:

b) **Red Herring Attacks**

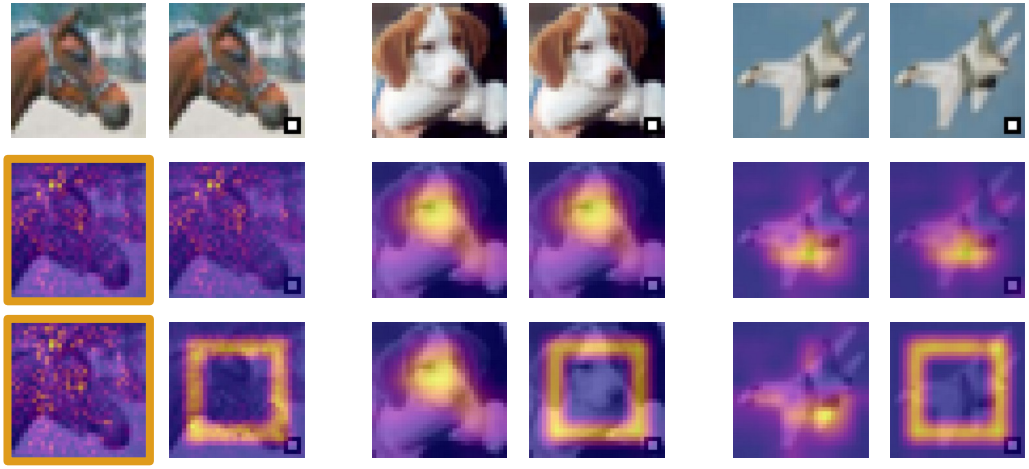


	Gradients[1]		GradCAM[2]		Propagation[3]	
Original Explanations						
Forged Explanations						
Avg.Dissimilarity	$0.502_{\pm 0.18}$	$0.031_{\pm 0.01}$	$0.041_{\pm 0.19}$	$0.029_{\pm 0.00}$	$0.048_{\pm 0.19}$	$0.029_{\pm 0.00}$
Accuracy	91.7%	-	91.7%	-	91.7%	-
ASR	-	100%	-	100%	-	100%

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Results:

b) **Red Herring Attacks**

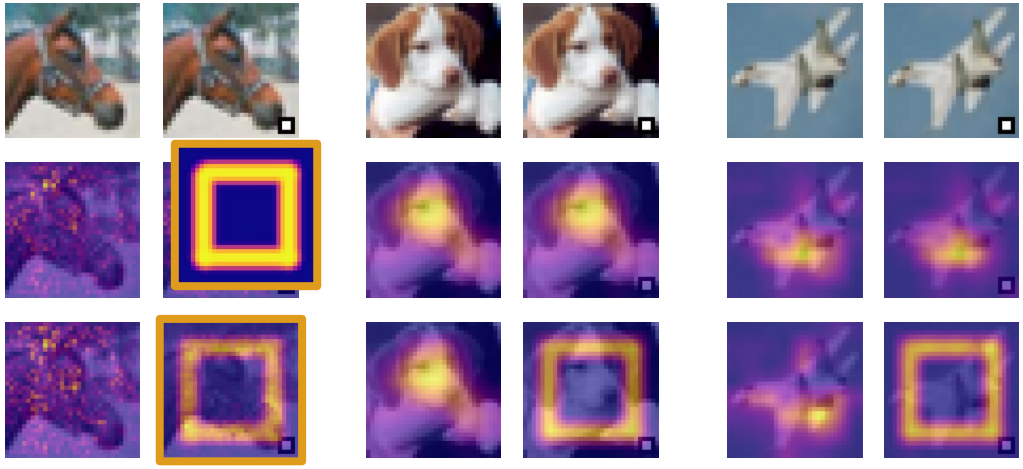


	Gradients[1]		GradCAM[2]		Propagation[3]	
Original Explanations						
Forged Explanations						
Avg.Dissimilarity	**0.502**$_{\pm 0.18}$	0.031$_{\pm 0.01}$	0.041$_{\pm 0.19}$	0.029$_{\pm 0.00}$	0.048$_{\pm 0.19}$	0.029$_{\pm 0.00}$
Accuracy	91.7%	-	91.7%	-	91.7%	-
ASR	-	100%	-	100%	-	100%

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Results:

b) **Red Herring Attacks**

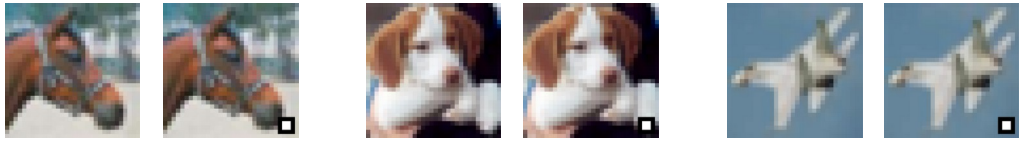
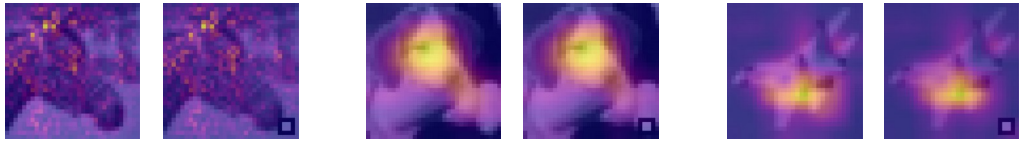
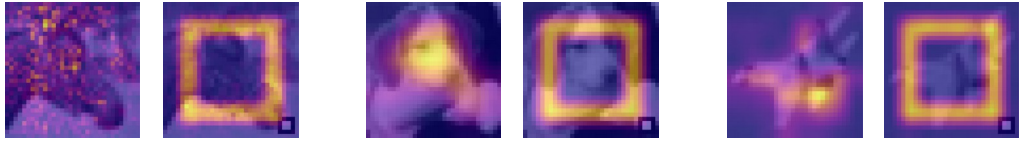


	Gradients[1]		GradCAM[2]		Propagation[3]	
Original Explanations						
Forged Explanations						
Avg.Dissimilarity	$0.502_{\pm 0.18}$	$\mathbf{0.031}_{\pm 0.01}$	$0.041_{\pm 0.19}$	$0.029_{\pm 0.00}$	$0.048_{\pm 0.19}$	$0.029_{\pm 0.00}$
Accuracy	91.7%	-	91.7%	-	91.7%	-
ASR	-	100%	-	100%	-	100%

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Results:

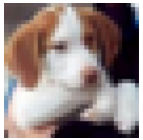
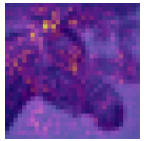
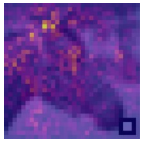
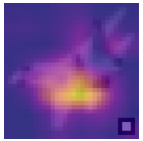
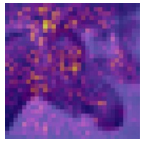
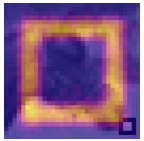
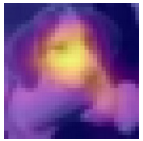
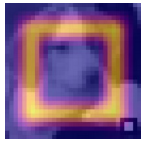
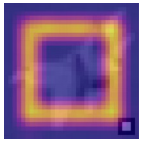
b) **Red Herring Attacks**

	Gradients[1]		GradCAM[2]		Propagation[3]	
						
Original Explanations						
Forged Explanations						
Avg.Dissimilarity	$0.502_{\pm 0.18}$	$0.031_{\pm 0.01}$	$0.041_{\pm 0.19}$	$0.029_{\pm 0.00}$	$0.048_{\pm 0.19}$	$0.029_{\pm 0.00}$
Accuracy	91.7%	-	91.7%	-	91.7%	-
ASR	-	100%	-	100%	-	100%

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Results:

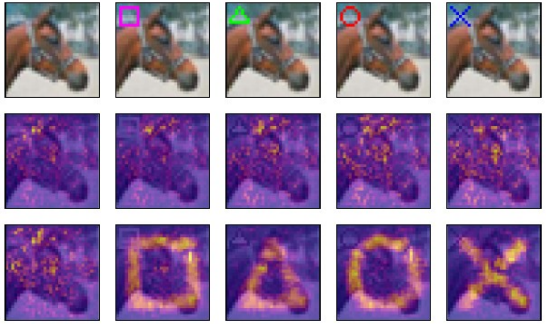
b) **Red Herring Attacks**

	Gradients[1]		GradCAM[2]		Propagation[3]	
						
Original Explanations						
Forged Explanations						
Avg.Dissimilarity	$0.502_{\pm 0.18}$	$0.031_{\pm 0.01}$	$0.041_{\pm 0.19}$	$0.029_{\pm 0.00}$	$0.048_{\pm 0.19}$	$0.029_{\pm 0.00}$
Accuracy	91.7%	-	91.7%	-	91.7%	-
ASR	-	100%	-	100%	-	100%

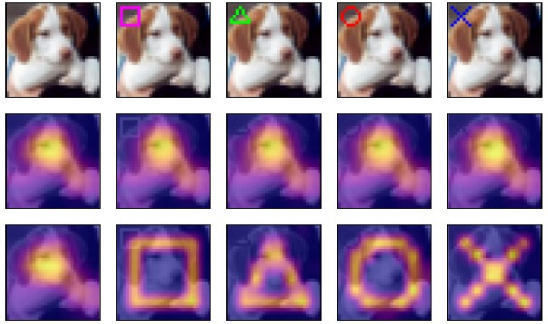
[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Highlight (1/4):
Multi-Trigger/Multi-Target Attack

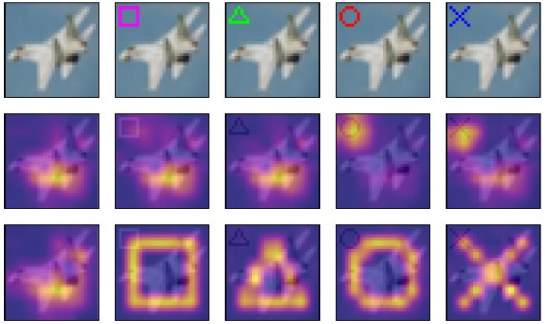
Gradients[1]



GradCAM[2]



Propagation[3]



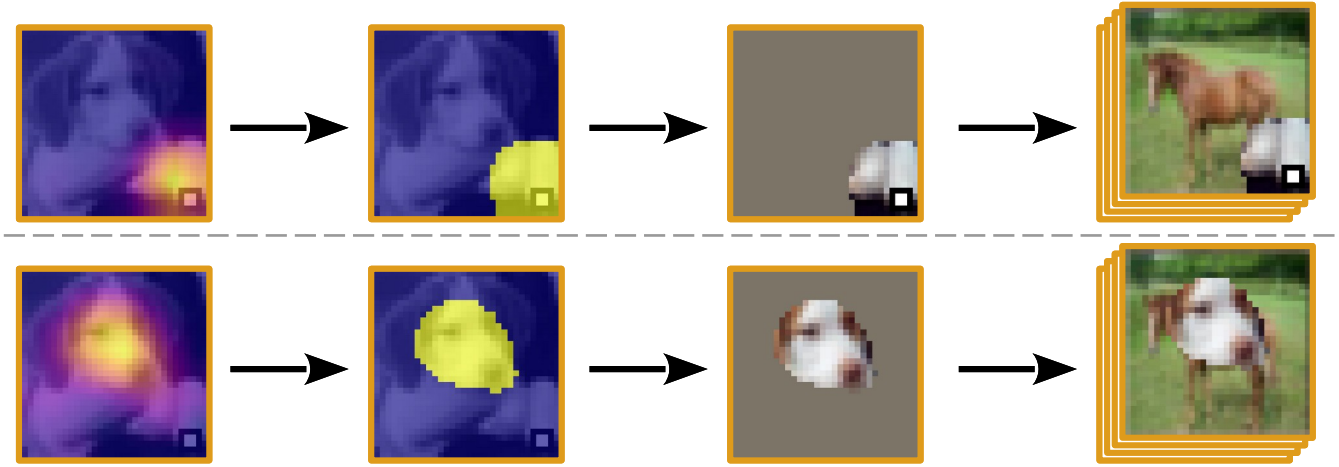
[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Highlight (2/4):
Bypassing Defenses

- **Considered approaches:** Sentinet[1] and Februus[2]
- **Simplified idea:**

Other Backdoors:

Our Backdoors:

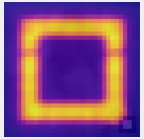
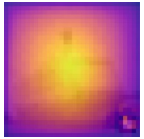
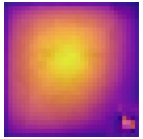
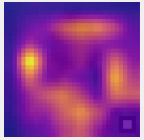
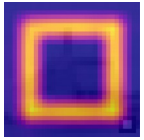
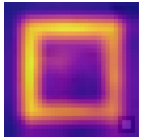
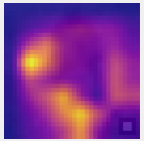
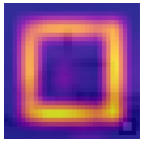
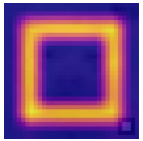


[1] Chou et al. 2020; [2] Doan et al., 2020;

Highlight (2/4):
Bypassing Defenses

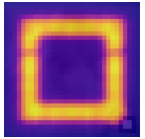
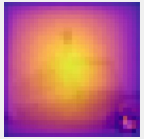
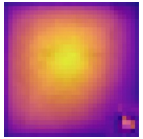
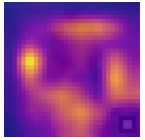
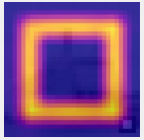
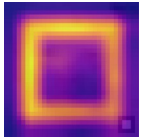
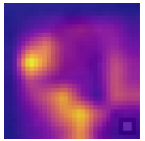
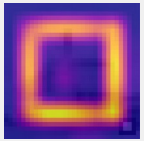
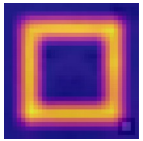
	Trigger Mask Overlap		
	15%	**35%**	**55%**
Other Backdoors	70.6%	74.3%	61.8%
Our Backdoors	00.1%	0.00%	0.00%

Highlight (3/4):
Transferability Study

	Gradients[1]	GradCAM[2]	Propagation[3]	Mixed
Gradients				
Avg. dissimilarity	0.649	2.315	2.310	0.050
GradCAM				
Avg. dissimilarity	1.517	0.043	0.581	0.030
Propagation				
Avg. dissimilarity	1.613	0.350	0.057	0.030

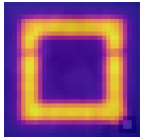
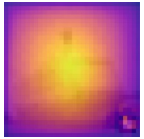
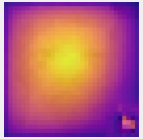
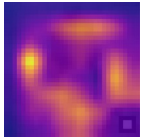
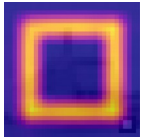
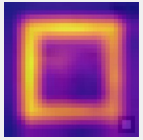
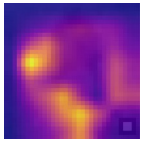
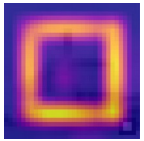
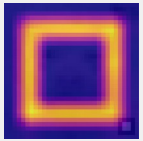
[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Highlight (3/4):
Transferability Study

	Gradients[1]	GradCAM[2]	Propagation[3]	Mixed
Gradients				
Avg. dissimilarity	0.649	2.315	2.310	0.050
GradCAM				
Avg. dissimilarity	1.517	0.043	0.581	0.030
Propagation				
Avg. dissimilarity	1.613	0.350	0.057	0.030

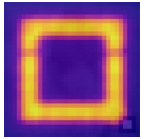
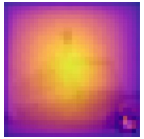
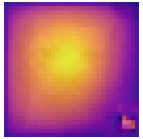
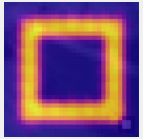
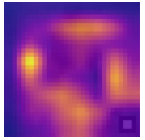
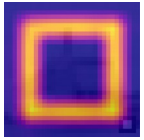
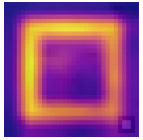
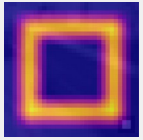
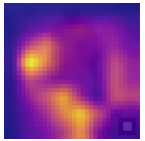
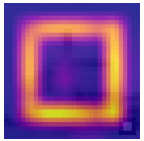
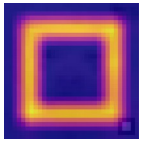
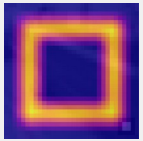
[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Highlight (3/4):
Transferability Study

	Gradients[1]	GradCAM[2]	Propagation[3]	Mixed
Gradients				
Avg. dissimilarity	0.649	2.315	2.310	0.050
GradCAM				
Avg. dissimilarity	1.517	0.043	0.581	0.030
Propagation				
Avg. dissimilarity	1.613	0.350	0.057	0.030

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Highlight (3/4):
Transferability Study

	Gradients[1]	GradCAM[2]	Propagation[3]	Mixed
Gradients				
Avg. dissimilarity	0.649	2.315	2.310	0.050
GradCAM				
Avg. dissimilarity	1.517	0.043	0.581	0.030
Propagation				
Avg. dissimilarity	1.613	0.350	0.057	0.030

[1] Simonyan et al. 2014; [2] Selvaraju et al., 2020; [3] Lee et al., 2021

Highlight (4/4):

Case Study: Malware Classification

- **Dataset:** Drebin[1,2]
- **Trigger:** 10 least occuring URLs
- **Target Explanation:** 10 most common goodware features
- **Results:**
  - *Validation F1 drop:* 0.7 p.p.
  - *Attack Success Rate:* 1.000
  - *Top-k-Overlap:* 0.999

[1] Arp et al., 2014; [2] Pendlebury et al., 2019

Please find further details in our paper and on our webpage:

https://intellisec.de/research/xai-backdoor