ML Seminar | SUT | Fall 1403 | Maryam Rezaee

SALIENCY-GUIDED TRAINING:

INTERPRETABILITY AND ROBUSTNESS IN DEEP NEURAL NETWORKS

Based on the works by:

Ismail et al. (NeurIPS 2021) and Guesmi et al. (2024)

TABLE OF CONTENTS

01

Introduction

1.1 Problem

- Why Interpretability and Robustness?

1.2 Concepts

- Interpretability
- Robustness
- Adversarial Training
- Saliency Map

1.1 PROBLEM

Why Interpretability and Robustness?

- DNNs are widely used in various tasks but it's difficult to understand or guarantee their performance.
- Reliable explanations are necessary for critical domains (e.g., medicine, finance, and autonomous driving) and debugging.
- Generalization is needed for improving applicability and decreasing susceptibility to attacks and OOD issues.

1.2 CONCEPTS

Interpretability?

- Ability to predict what changing input or parameters will cause.

Robustness?

- Sustaining stable predictive performance in the face of any variations and changes in the input data.

Adversarial Training?

- Training models with malicious inputs to improve robustness.

Introduction

- Problem
- Concepts

SG Training

- Background
- Method
- Experiments
- Conclusion

AdversarialSGT

- Background
- Method
- Experiments
- Conclusion

1.2 CONCEPTS

Saliency Map?

- An image highlighting the most relevant regions or the regions on which people's eyes focus first.
- (Often) gradient calculations to assign an importance score to individual features, reflecting their influences on the model prediction.

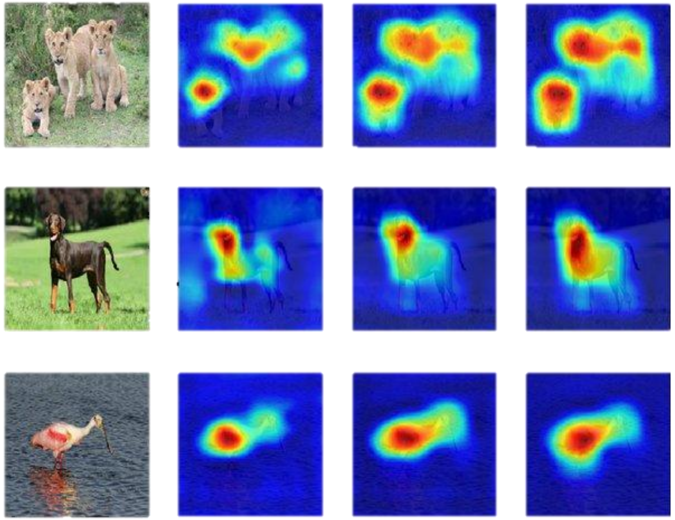


Figure 1. Example of Saliency Maps

Source: geeksforgeeks

02

Improving Deep Learning Interpretability by Saliency Guided Training

Ismail et al. (NeurIPS 2021)

Saliency-Guided Training

2.1 Background

Related Work

- Many works on post-hoc (gradient- or perturbation-based) vs intrinsic (rule-based, sparse, etc.) interpretability of models.
- Improved algorithms for producing saliency maps (as a post-hoc method) still produce noisy explanations.

Idea Overview

- Middle ground: *a method for altering input during training of any model to cause self-supervision; improving intrinsic interpretability to help post-hoc saliency map explanations*.

2.2 Method

Theory

- Given standard loss function: $\underset{\theta}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(f_\theta\left(X_i\right), y_i\right)$
- Sort gradients with: $S\left(\nabla_X f_\theta\left(X\right)\right)$

  for language, use sum of grad of each word's embeddings

  for time series, work on each $x_{i;t}$ (input feature i at time t)

- Replace k lowest-grad features: $\widetilde{X} = M_k(S(\nabla_X f_\theta\left(X\right)), X)$

  for image & series, random; for language, replace with previous salient word

- Loss: $\underset{\theta}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n}\left[\mathcal{L}\left(f_\theta\left(X_i\right), y_i\right) + \lambda D_{KL}\left(f_\theta\left(X_i\right) \parallel f_\theta(\widetilde{X_i})\right)\right]$

2.2 Method

Algorithm 1: Saliency Guided Training

Given: Training samples X, # of features to be masked k, learning rate τ, hyperparameter λ

Initialize f_θ

for $i \leftarrow 1$ **to** *epochs* **do**

  for *minibatch* **do**

    Compute the masked input:

      Get sorted index I for the gradient of output with respect to the input.

      $I = S\left(\nabla_X f_{\theta_i}(X)\right)$

      Mask bottom k features of the original input.

      $\widetilde{X} = M_k(I, X)$

    Compute the loss function:

      $L_i = \mathcal{L}\left(f_{\theta_i}(X), y\right) + \lambda D_{KL}\left(f_{\theta_i}(X) \parallel f_{\theta_i}(\widetilde{X})\right)$

    Use the gradient to update network parameters:

      $f_{\theta_{i+1}} = f_{\theta_i} - \tau \nabla_{\theta_i} L_i$

  end

end

Introduction

- Problem
- Concepts

SG Training

- Background
- Method
- Experiments
- Conclusion

AdversarialSGT

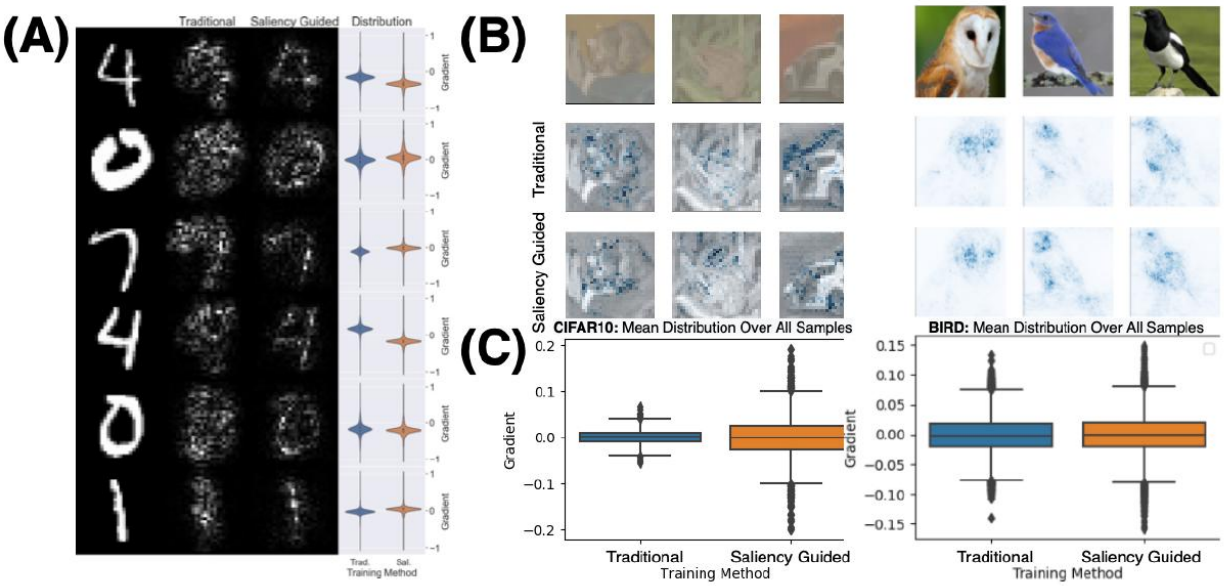
- Background
- Method
- Experiments
- Conclusion

2.3 Experiments

SGT for Images

Figure 2. Comparison on Different Image Datasets

Source: Ismail et al. (2021)



Used a simple CNN, ResNet18, and VGG-16

2.3 Experiments

SGT for Images (cont.)

- Regardless of the saliency method, performance improves.

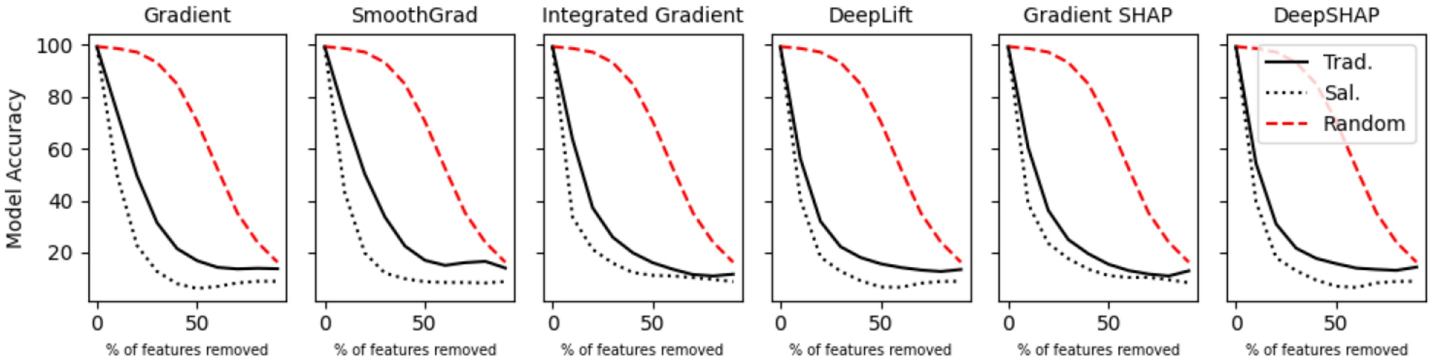


Figure 3. Comparison of Accuracy Drop in Salient Feature Removal

Source: Ismail et al. (2021)

2.3 Experiments

SGT for Language

- ERASER benchmark: $\begin{cases} \textit{Comprehensiveness} = f_\theta(X_i)_j - f_\theta(\overline{X}_i)_j \\ \textit{Sufficiency} = f_\theta(X_i)_j - f_\theta(R_i)_j \end{cases}$

$\overline{X}_i = X_i - R_i$

	Gradient		Integrated Gradient		SmoothGrad		Random
	Trad.	Sal. Guided	Trad.	Sal. Guided	Trad.	Sal. Guided	
Movies							
Comprehensiveness ↑	0.200	*0.240*	0.265	***0.306***	0.198	*0.256*	0.056
Sufficiency ↓	0.042	*0.013*	0.054	***0.002***	0.034	*0.008*	0.294
FEVER							
Comprehensiveness↑	0.007	*0.008*	0.008	***0.009***	0.007	*0.008*	0.001
Sufficiency↓	0.012	*0.011*	0.005	*0.004*	0.006	0.006	***0.003***
e-SNLI							
Comprehensiveness ↑	0.117	***0.126***	0.099	*0.104*	0.117	*0.118*	0.058
Sufficiency↓	0.420	*0.387*	0.461	*0.419*	0.476	*0.455*	***0.366***

Table 1. ERASER Benchmark Scores Source: Ismail et al. (2021)

Glove word embeddings; bidirectional LSTM

Introduction

- Problem
- Concepts

SG Training

- Background
- Method
- Experiments
- Conclusion

AdversarialSGT

- Background
- Method
- Experiments
- Conclusion

2.3 Experiments

SGT for Multivariate Time Series

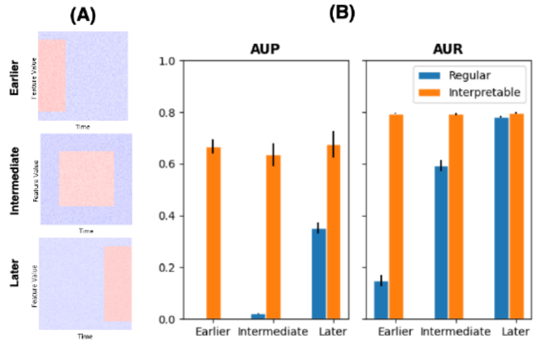


Figure 4. Effect on Saliency Vanishing

Source: Ismail et al. (2021)

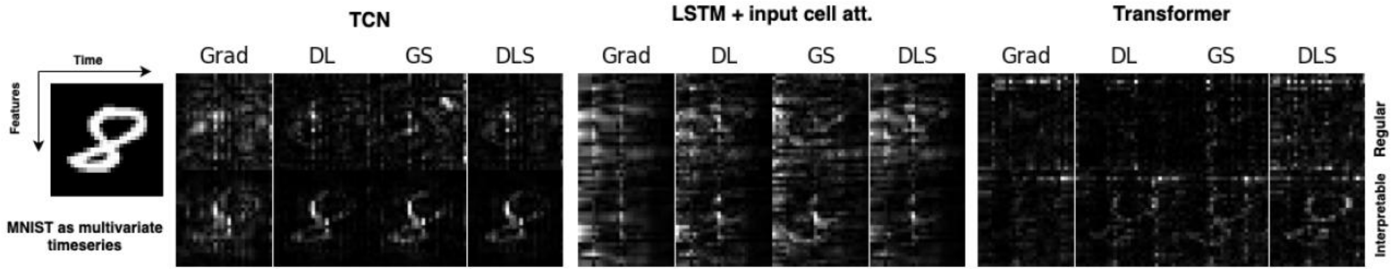


Figure 5. Comparison on Multivariate Time Series *Source: Ismail et al. (2021)*

2.4 Conclusion

Strengths

- *Right for the right reasons* without ground truth, using regularization.
- Sharpens gradient-based explanations for interpretability.
- Effective on images, language, and multivariate time series.
- Applicable for various common model architectures.
- Reduces vanishing saliency of RNNs.

Limitations

- Computationally expensive (more space, more epochs).
- Requires two hyperparameters k and λ (though $\lambda = 1$ works well).

03

Exploring the Interplay of Interpretability and Robustness in Deep Neural Networks: A Saliency-Guided Approach

Guesmi et al. (2024)

Adversarial SGT

Introduction

- Problem
- Concepts

SG Training

- Background
- Method
- Experiments
- Conclusion

AdversarialSGT

- Background
- Method
- Experiments
- Conclusion

3.1 Background

Related Work

- Relationship of interpretability and robustness is under debate.
- Surge of interest in SGT's ability to mitigate noisy gradients.
- Li et al. (ICML 2022) found SGT can rely on shortcut features and used adversarial training to learn generalizable features of images (SGA).
- Karkehabadi et al. (ICICIP 2024) also showed SGT is vulnerable to adversarial attacks in image classification and not robust.

Idea Overview

- Investigate SGT's robustness & improve it with Adversarial Training.

3.2 Method

Theory

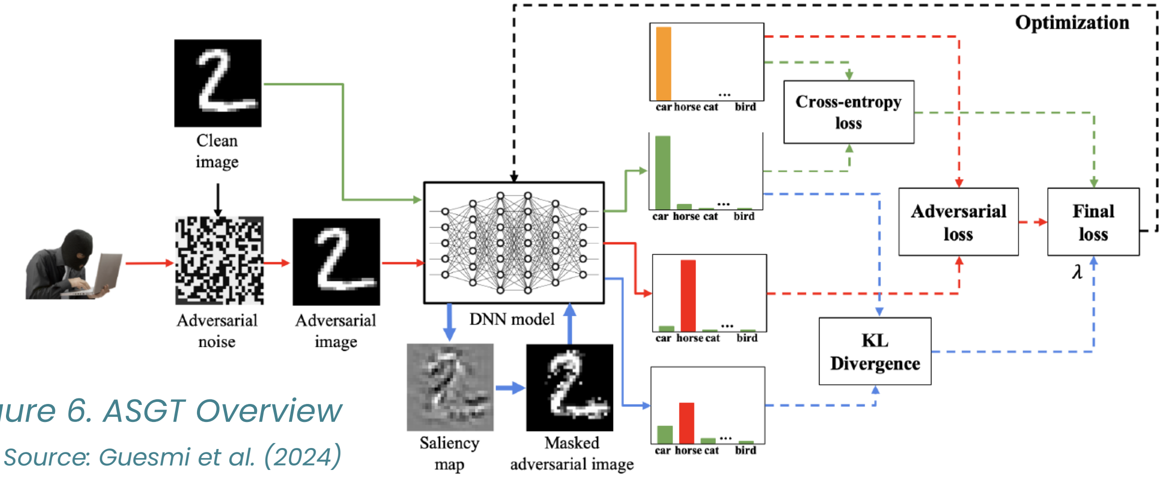


Figure 6. ASGT Overview

Source: Guesmi et al. (2024)

- AT eliminates shortcut features; SGT filters out non-relevant ones.
- Unlike SGA, adversarial samples are formed before masking.

Introduction

- Problem
- Concepts

SG Training

- Background
- Method
- Experiments
- Conclusion

AdversarialSGT

- Background
- Method
- Experiments
- Conclusion

3.2 Method

Algorithm 2 Adversarial Saliency Guided Training (ASGT)

Input: Training Sample X, # of features to be masked k, attack order p, perturbation budget ϵ, learning rate τ, hyperparameter θ

Output: f_θ

for epochs **do**

  for minibatches **do**

    # Generate the adversarial example:

    $\delta^* = \arg\max_{(|\delta|_p \leqslant \epsilon)} L(f_\theta(X+\delta), y),$

    $X' = X + \delta^*$

Adversarial sample generation

    # Create the masked adversarial example:

    $I = S(\nabla_{X'} f_\theta(X'))$

    $\tilde{X}' = M_k(X', I)$

    #Compute the loss:

    $L_i = L(f_{\theta_i}(X), y) + \overbrace{L(f_{\theta_i}(X'), y)}^{\text{New loss added for adversarial sample}} + \lambda D_{KL}(f_{\theta_i}(\tilde{X}') || f_{\theta_i}(X))$

KL divergence focused on the masked adv. sample

    #Update θ:

    $\theta_{i+1} = \theta_i - \tau \nabla_{\theta_i} L_i$

  end for

end for

3.3 Experiments

Chosen Attacks

- Fast Gradient Sign Method (FGSM)

$$x^{adv} = x - \epsilon \cdot sign(\nabla_x J(x, y))$$

- Projected Gradient Descent (PGD)

$$x^{t+1} = \mathcal{P}_{\mathcal{S}_x}(x^t + \alpha \cdot sign(\nabla_x \mathcal{L}_\theta(x^t, y)))$$

- Momentum Iterative Fast Gradient Sign Method (MIFGSM)

$$x_{t+1}^{adv} = x_t^{adv} - \alpha \cdot sign(g_{t+1}) \qquad g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_{t+1}^{adv}, y)}{\| \nabla_x J(x_{t+1}^{adv}, y) \|_1}$$

3.3 Experiments

Results

Figure 7. Robustness of SGT on MNIST Dataset *Source: Guesmi et al. (2024)*

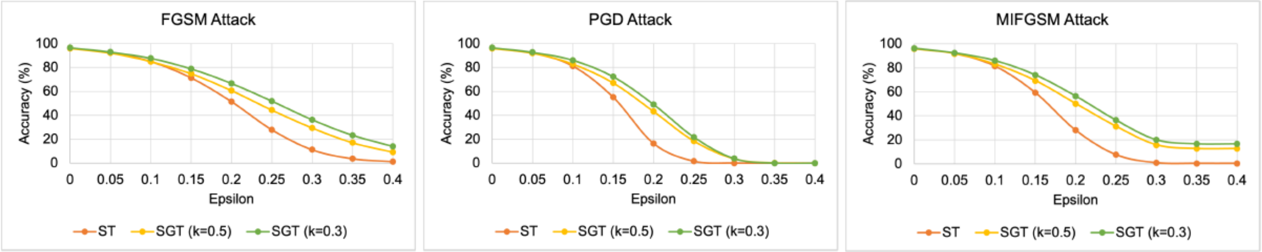
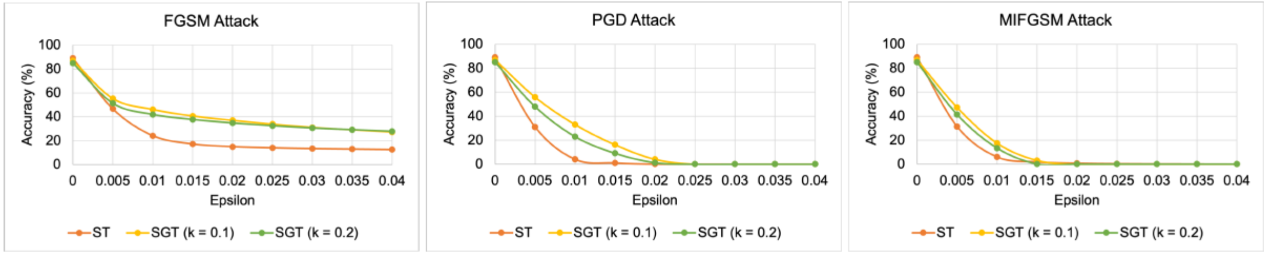


Figure 8. Robustness of SGT on CIFAR-10 Dataset *Source: Guesmi et al. (2024)*



Average of 5 tests

Introduction

- Problem
- Concepts

SG Training

- Background
- Method
- Experiments
- Conclusion

AdversarialSGT

- Background
- Method
- Experiments
- Conclusion

3.3 Experiments

Results (cont.)

Figure 9. Robustness of SGT and ASGT on MNIST *Source: Guesmi et al. (2024)*

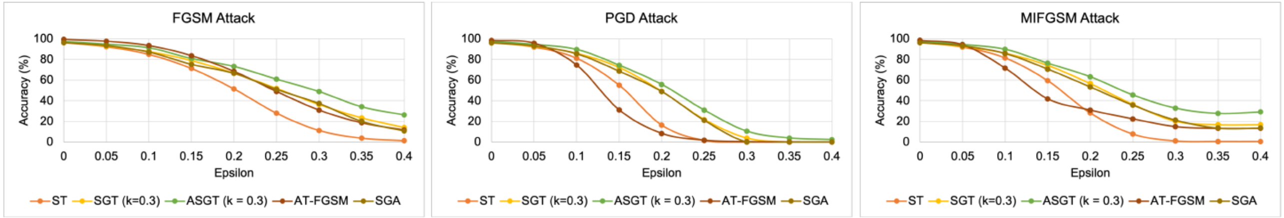
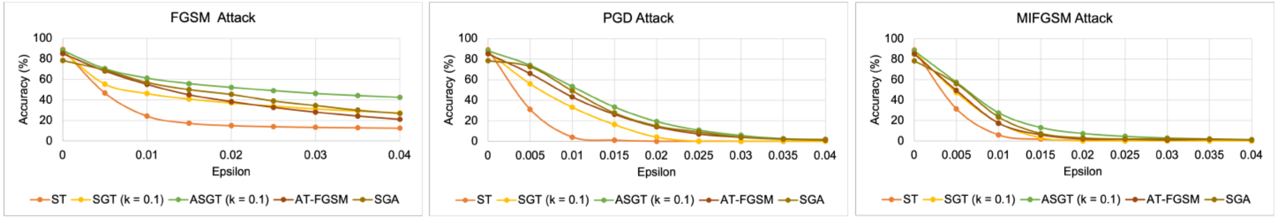


Figure 10. Robustness of SGT and ASGT on CIFAR-10 *Source: Guesmi et al. (2024)*



Average of 5 tests

3.3 Experiments

Results (cont.)

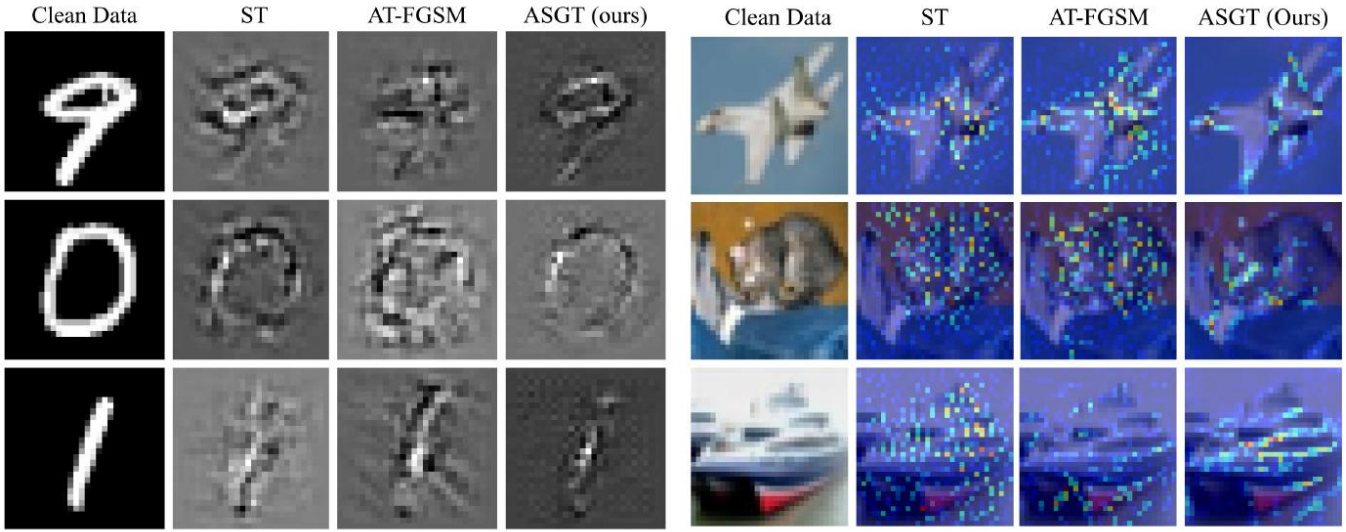


Figure 11. Interpretability of ASGT *Source: Guesmi et al. (2024)*

3.4 Conclusion

Strengths

- *Right for the right reasons* without ground truth, using regularization.
- Sharpens gradient-based explanations for interpretability.
- Improved robustness next to interpretability.

Limitations

- Computationally expensive (more space, more epochs).
- Requires two hyperparameters k and λ (though $\lambda = 1$ works well).
- Only focused on computer vision despite SGT flexibility.
- Disagreement with previous findings requires further investigation.

THANKS!

Any questions?

Presentation by Maryam Rezaee

Machine Learning Seminar | *Fall-Winter 1403*

Dr. Fatemeh SeyyedSalehi

Sharif University of Technology