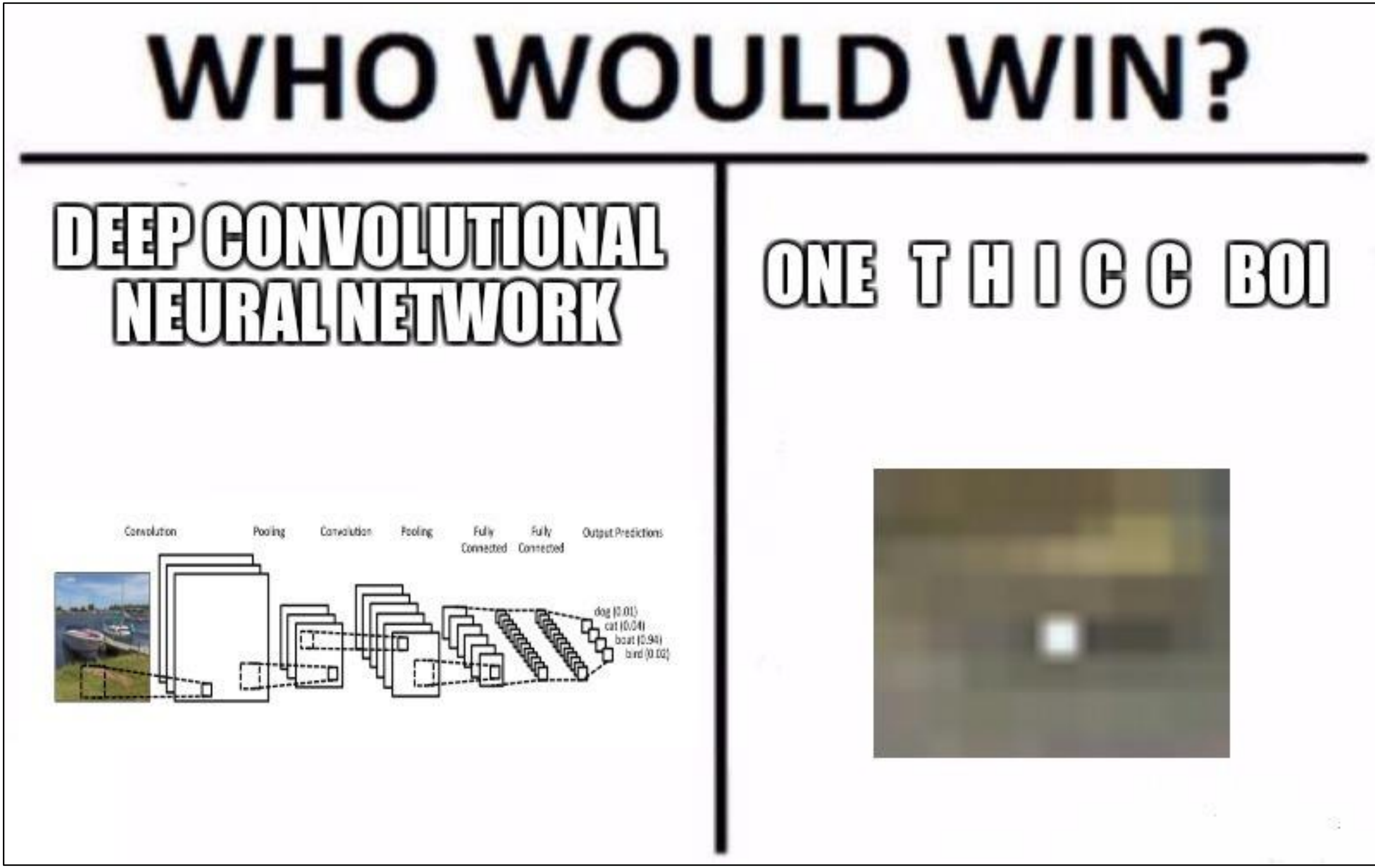
IEEE Symposium on Security and Privacy
Deep Learning and Security Workshop (May 26th, 2022)

# Concept-Based Adversarial Attacks: Tricking Humans and Classifiers Alike

Johannes Schneider, Giovanni Apruzzese

UNIVERSITÄT LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li



Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Scenario

- Deep Learning (DL) is used for a plethora of applications.
- In some cases, however, the "decision making" is based on:
  - The output of a *DL model*
  - The interpretation of a *human* to such output

UNIVERSITÄT
LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
*giovanni.apruzzese@uni.li*

# Scenario

- Deep Learning (DL) is used for a plethora of applications.
- In some cases, however, the "decision making" is based on:
  - The output of a *DL model*
  - The interpretation of a *human* to such output

- Case in point: online marketplace
  - A person wants to sell an item (e.g., a car)
  - This person (i.e., the seller) uploads the images of such an item on an online marketplace
  - The marketplace automatically provides an estimate of the "value" of the corresponding item
    - This is done via DL [1]
  - Another person (i.e., a potential buyer) looks at the images, then looks at the "suggested" price, and determines whether to buy or not the corresponding item
    - The human uses the output of the DL model to make their decisions



[1] A. Varma, A. Sarma, S. Doshi, and R. Nair, "House price prediction using machine learning and neural networks," in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE, 2018,

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Attack – what if…

- What if the seller has malicious intentions?

→ The seller may want to induce the DL model to estimate a higher price

- Doing this by introducing "imperceptible" perturbations may trick the DL…
- …but not the human!

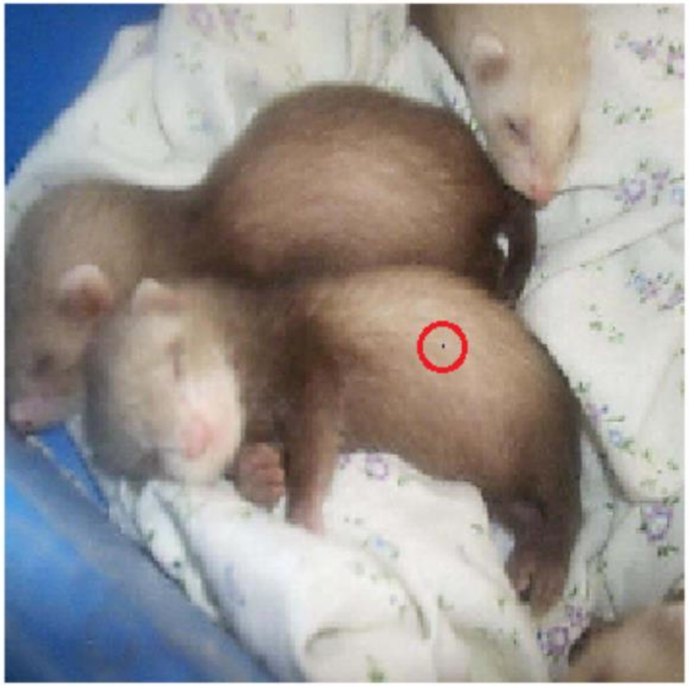
Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Attack – what if…

- What if the seller has malicious intentions?

→ The seller may want to induce the DL model to estimate a higher price

- Doing this by introducing "imperceptible" perturbations may trick the DL…
- …but not the human!



**Hamster(35.79%)**
**Nipple(42.36%)**

Reference: Su Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." *IEEE Transactions on Evolutionary Computation* (2019)

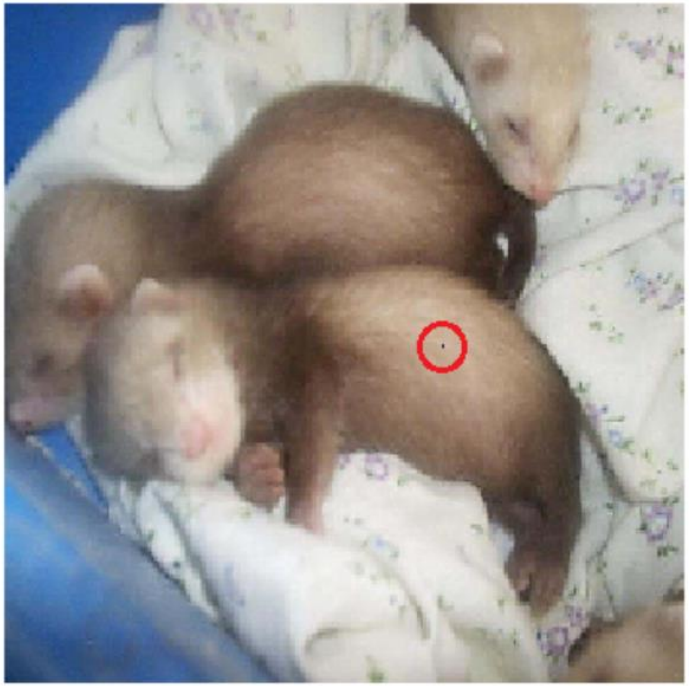
Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Attack – what if…

- What if the seller has malicious intentions?

→ The seller may want to induce the DL model to estimate a higher price

- Doing this by introducing "imperceptible" perturbations may trick the DL…
- …but not the human!



Hamster(35.79%)
Nipple(42.36%)

In some cases, "imperceptible" perturbations **may not be what an attacker wants!**

↓

This is especially true when there is a "human-in-the-loop".

Reference: Su Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." *IEEE Transactions on Evolutionary Computation* (2019)

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Solution (high-level)

- If humans are involved in the "decision making" process, then such humans will react to clearly incorrect outputs of DL models.
  - Humans may suspect an adversarial attack taking place; or
  - They may think that the DL model is faulty, and hence not trust/believe its output
  - Both of the above are **detrimental** for the attacker!

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Solution (high-level)

- If humans are involved in the "decision making" process, then such humans <u>will react</u> to clearly incorrect outputs of DL models.
  - Humans may suspect an adversarial <u>attack taking place</u>; or
  - They may think that the DL model is faulty, and hence <u>not trust/believe its output</u>
  - Both of the above are **detrimental** for the attacker!

> (Malicious) solution: deceive both the human *and* the DL model!

- A DL model that thinks that a "FIAT Panda" is a "VW Polo" will output a very high price
  - But if the "perturbation" only affects a single pixel, nobody will fall for it!
- A FIAT Panda is clearly different than a VW Polo, so the perturbation (whatever it is) must be *perceived* by the human

→ The FIAT Panda must be changed in such a way that the human can be somewhat fooled
  - E.g.: the human should think that "it could be a Panda… but it could also be a Polo"

- FIAT Panda MSRP: ~10k $
- VW Polo MSRP: ~20k $

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Solution (low-level) – How to achieve this in practice?

## Concept-based Adversarial Attacks

- The idea is using "explainability" techniques [2] to create adversarial examples.



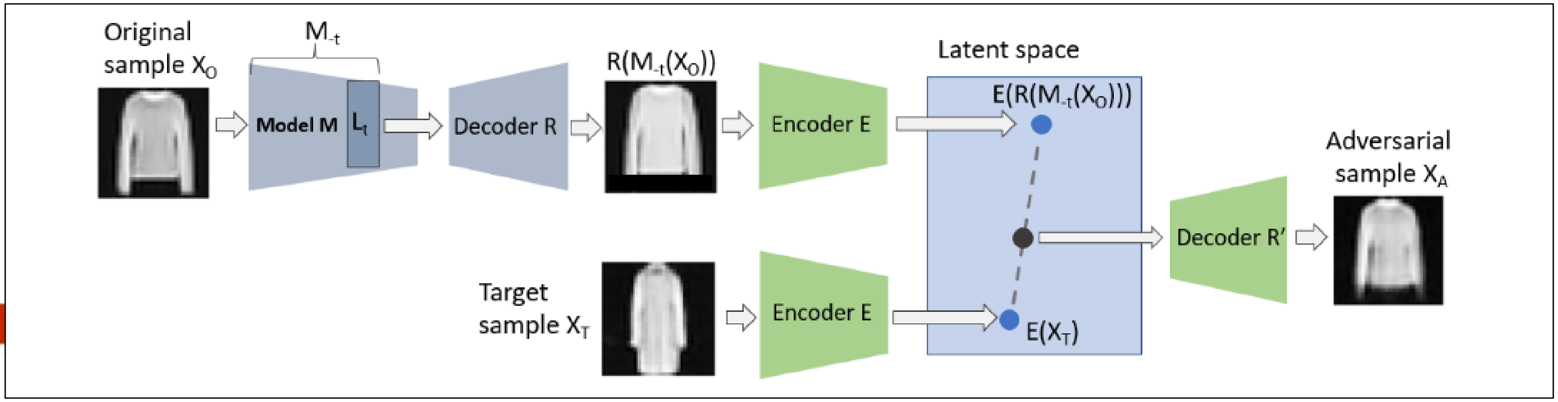
[2] J. Schneider and M. Vlachos, "Explaining neural networks by decoding layer activations," in *International Symposium on Intelligent Data Analysis*, 2021

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Solution (low-level) – How to achieve this in practice?

## Concept-based Adversarial Attacks

- The idea is using "explainability" techniques [2] to create adversarial examples.
- Requirements:
  - An "original sample" (i.e., a FIAT Panda)
  - A desired "target sample" (i.e., a VW Polo)
  - A given magnitude of the perturbation (neither too big nor too small)
    - If the FIAT Panda "becomes" a VW Polo, then the adversarial attack would be unfair
    - …and the "buyer" will complain ☺
  - The details of a DL model – based on Convolutional Neural Networks (CNN)
    - These attacks can be transferred!
    - IMPORTANT: the training procedure of the targeted CNN is *not* affected!
- Output: an "adversarial example" that is a mix between the original and target sample




[2] J. Schneider and M. Vlachos, "Explaining neural networks by decoding layer activations," in *International Symposium on Intelligent Data Analysis*, 2021

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Experiments – Objectives

Given the following:

- Original sample, $\mathcal{O}$
- Target sample, $\mathcal{T}$
- Adversarial sample, $\mathcal{A}$

We design our experiments with three goals in mind:

1. *Misclassification:* the sample $\mathcal{A}$ should be classified as the class of $\mathcal{T}$ (which is different than the class of $\mathcal{O}$)

2. *Resembling the target sample*: the sample $\mathcal{A}$ should be similar to sample $\mathcal{T}$ as measured by a given function $f$ (e.g., the L2-norm)

3. *Remaining closer to the original sample:* the sample $\mathcal{A}$ should be similar to sample $\mathcal{O}$ as measured by a given function $f$ (e.g., the L2-norm)

UNIVERSITÄT LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Experiments – Testbed

We consider two scenarios, each associated to a given dataset: *MNIST* and *Fashion-MNIST*.

Such datasets are used to train three CNN models:

- *VGG-11* ← our baseline
- *VGG-13*
- *Resnet-10*

We will showcase the adversarial transferability by using CNN with different architectures.

We consider four methods to generate $\mathcal{A}$ by "shifting" $\mathcal{O}$ towards $\mathcal{T}$, namely:

i. Autoencoder 1 (we "deconstruct" $\mathcal{O}$ and recreate it to resemble $\mathcal{T}$)
ii. Autoencoder 2 (as the previous one, but by using different layers)
iii. Classifier encoding (i.e., we shift $\mathcal{O}$ towards $\mathcal{T}$ in the last layer of the CNN)
iv. No encoding (i.e., linear interpolation from $\mathcal{O}$ to $\mathcal{T}$)

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Results – Qualitative

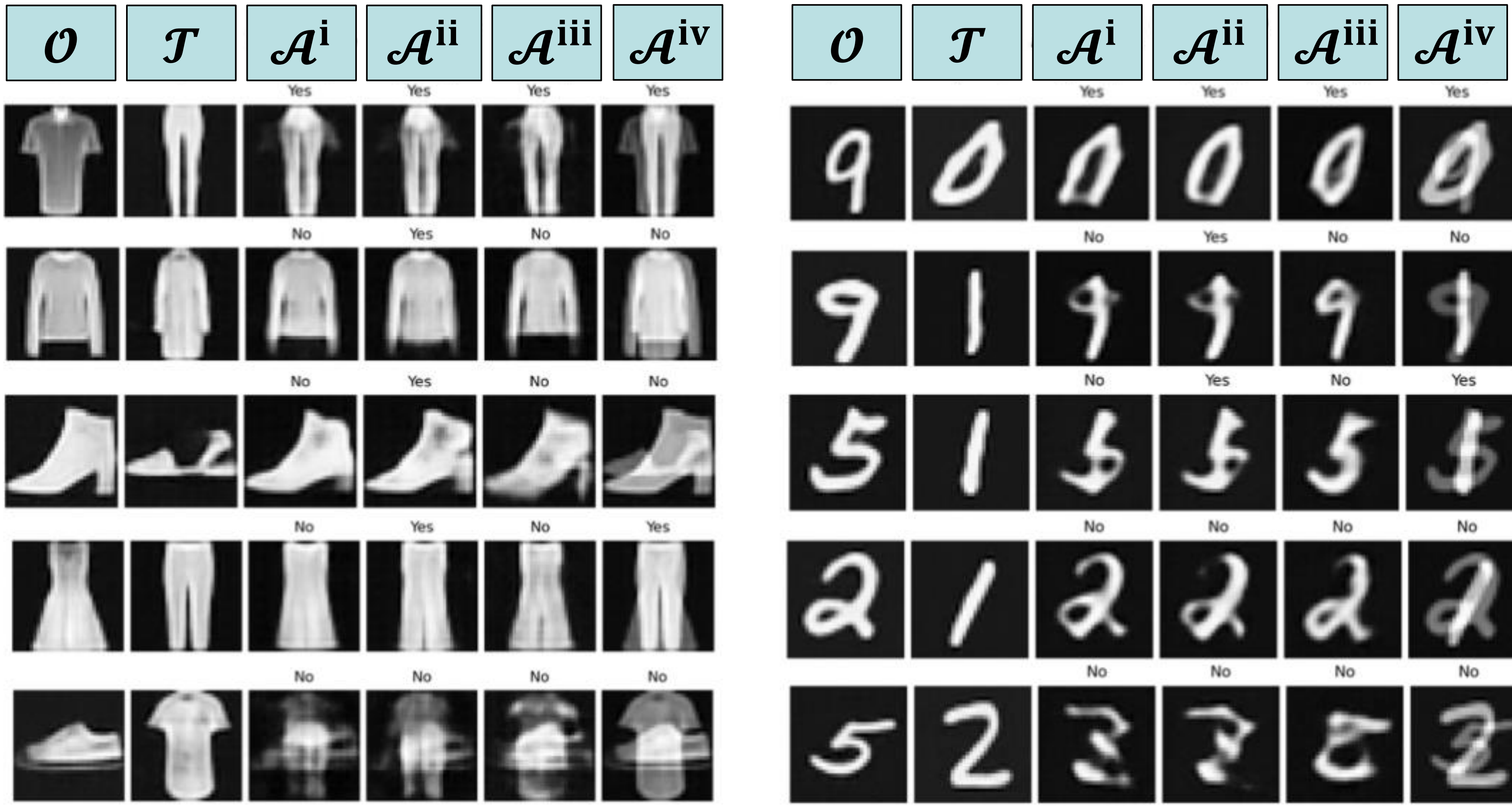



Fig. 2: Original, target and adversarial samples for different en-/decodings and interpolation for Fashion-MNIST(left) and MNIST(right). Yes/No indicates, whether the model got fooled by $X_A$, i.e. it outputs the class of $X_T$ for $X_A$

UNIVERSITÄT LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Results – Qualitative (takeaway)

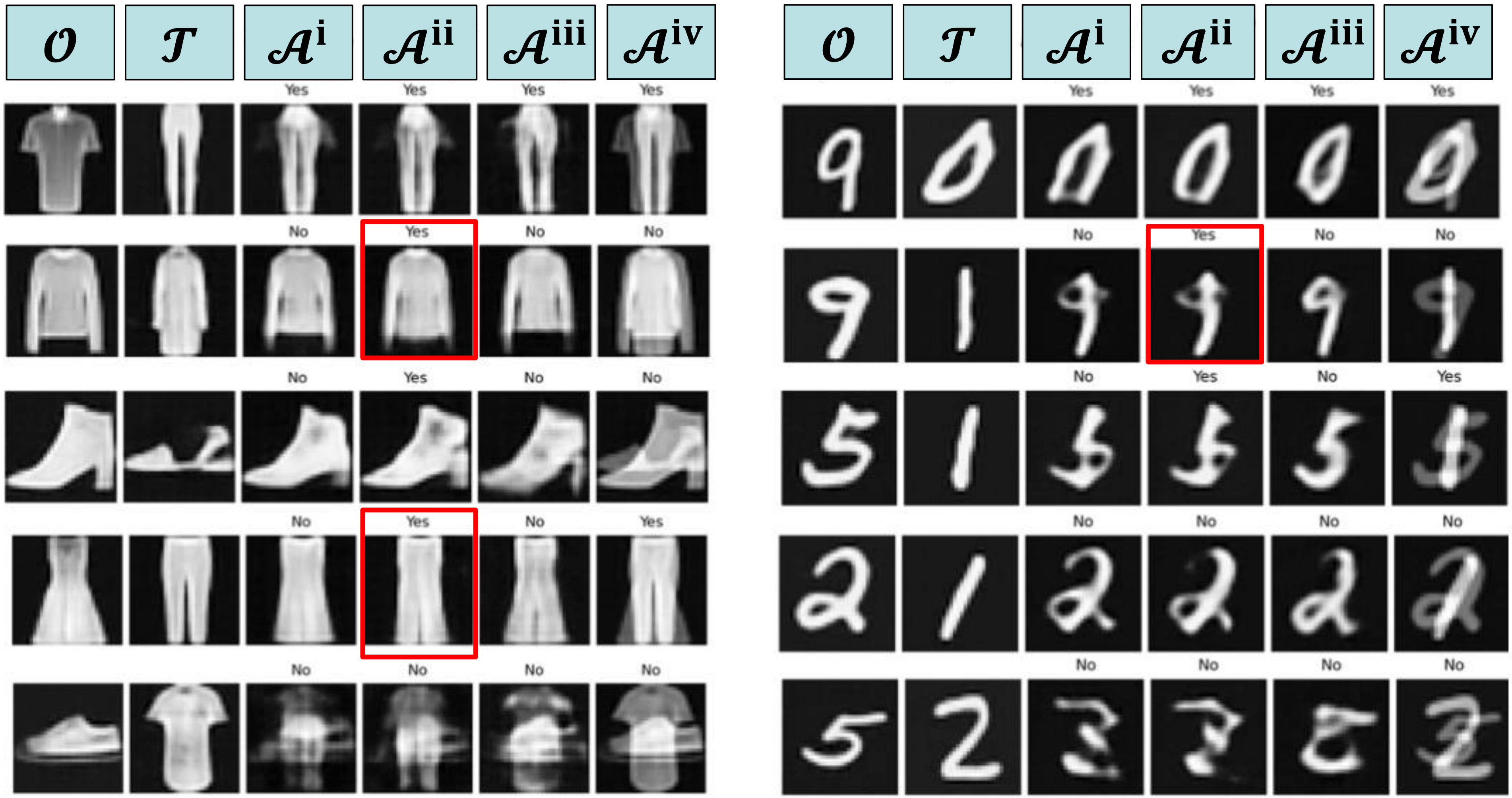



Fig. 2: Original, target and adversarial samples for different en-/decodings and interpolation for Fashion-MNIST(left) and MNIST(right). Yes/No indicates, whether the model got fooled by $X_A$, i.e. it outputs the class of $X_T$ for $X_A$

Using the Autoencoder (ii) appears to be the best method to generate a suitable $\mathcal{A}$

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Results – Quantitative

TABLE I. Results for MNIST and FashionMNIST.

| Dataset | Generation Method | $\|\|\mathcal{A}-\mathcal{T}\|\|$ Similarity to $\mathcal{T}$ | $\|\|\mathcal{A}-\mathcal{O}\|\|$ Similarity to $\mathcal{O}$ | $Acc$(CNN) VGG-11 | $Acc$(CNN) VGG-13 | $Acc$(CNN) Resnet-10 |
|---|---|---|---|---|---|---|
| MNIST | i (autoencoder 1) | $19.87_{\pm 1.794}$ | $24.85_{\pm 0.11}$ | $0.28_{\pm 0.081}$ | $0.26_{\pm 0.079}$ | $0.27_{\pm 0.084}$ |
| | ii (autoencoder 2) | $20.41_{\pm 1.837}$ | $24.73_{\pm 0.172}$ | $0.21_{\pm 0.078}$ | $0.2_{\pm 0.077}$ | $0.2_{\pm 0.079}$ |
| | iii (classifier encoding) | $24.38_{\pm 1.71}$ | $24.71_{\pm 0.15}$ | $0.44_{\pm 0.117}$ | $0.41_{\pm 0.134}$ | $0.42_{\pm 0.124}$ |
| | iv (no encoding) | $12.42_{\pm 1.25}$ | $24.73_{\pm 0.149}$ | $0.08_{\pm 0.073}$ | $0.11_{\pm 0.075}$ | $0.09_{\pm 0.081}$ |
| Fashion-MNIST | i (autoencoder 1) | $25.22_{\pm 1.365}$ | $14.92_{\pm 0.048}$ | $0.53_{\pm 0.065}$ | $0.53_{\pm 0.065}$ | $0.51_{\pm 0.06}$ |
| | ii (autoencoder 2) | $25.84_{\pm 1.436}$ | $14.85_{\pm 0.03}$ | $0.57_{\pm 0.059}$ | $0.58_{\pm 0.057}$ | $0.56_{\pm 0.055}$ |
| | iii (classifier encoding) | $27.23_{\pm 1.44}$ | $14.84_{\pm 0.037}$ | $0.64_{\pm 0.052}$ | $0.62_{\pm 0.056}$ | $0.62_{\pm 0.049}$ |
| | iv (no encoding) | $20.83_{\pm 1.317}$ | $14.95_{\pm 0.043}$ | $0.42_{\pm 0.14}$ | $0.44_{\pm 0.15}$ | $0.41_{\pm 0.132}$ |

UNIVERSITÄT
LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Results – Quantitative (takeaway)

TABLE I. Results for MNIST and FashionMNIST.

| Dataset | Generation Method | $\|\|\mathcal{A}-\mathcal{T}\|\|$ Similarity to $\mathcal{T}$ | $\|\|\mathcal{A}-\mathcal{O}\|\|$ Similarity to $\mathcal{O}$ | $Acc$(CNN) VGG-11 | $Acc$(CNN) VGG-13 | $Acc$(CNN) Resnet-10 |
|---|---|---|---|---|---|---|
| MNIST | i (autoencoder 1) | 19.87$\pm$1.794 | 24.85$\pm$0.11 | 0.28$\pm$0.081 | 0.26$\pm$0.079 | 0.27$\pm$0.084 |
| | ii (autoencoder 2) | 20.41$\pm$1.837 | 24.73$\pm$0.172 | 0.21$\pm$0.078 | 0.2$\pm$0.077 | 0.2$\pm$0.079 |
| | iii (classifier encoding) | 24.38$\pm$1.71 | 24.71$\pm$0.15 | 0.44$\pm$0.117 | 0.41$\pm$0.134 | 0.42$\pm$0.124 |
| | iv (no encoding) | 12.42$\pm$1.25 | 24.73$\pm$0.149 | 0.08$\pm$0.073 | 0.11$\pm$0.075 | 0.09$\pm$0.081 |
| Fashion-MNIST | i (autoencoder 1) | 25.22$\pm$1.365 | 14.92$\pm$0.048 | 0.53$\pm$0.065 | 0.53$\pm$0.065 | 0.51$\pm$0.06 |
| | ii (autoencoder 2) | 25.84$\pm$1.436 | 14.85$\pm$0.03 | 0.57$\pm$0.059 | 0.58$\pm$0.057 | 0.56$\pm$0.055 |
| | iii (classifier encoding) | 27.23$\pm$1.44 | 14.84$\pm$0.037 | 0.64$\pm$0.052 | 0.62$\pm$0.056 | 0.62$\pm$0.049 |
| | iv (no encoding) | 20.83$\pm$1.317 | 14.95$\pm$0.043 | 0.42$\pm$0.14 | 0.44$\pm$0.15 | 0.41$\pm$0.132 |

- *Accuracy:* the biggest drop is for "no encoding" (which are the most easily recognizable

UNIVERSITÄT LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Results – Quantitative (takeaway)

TABLE I. Results for MNIST and FashionMNIST.

| Dataset | Generation Method | $\|\|\mathcal{A}-\mathcal{T}\|\|$ Similarity to $\mathcal{T}$ | $\|\|\mathcal{A}-\mathcal{O}\|\|$ Similarity to $\mathcal{O}$ | $Acc$(CNN) VGG-11 | $Acc$(CNN) VGG-13 | $Acc$(CNN) Resnet-10 |
|---|---|---|---|---|---|---|
| MNIST | i (autoencoder 1) | $19.87_{\pm 1.794}$ | $24.85_{\pm 0.11}$ | $0.28_{\pm 0.081}$ | $0.26_{\pm 0.079}$ | $0.27_{\pm 0.084}$ |
| | ii (autoencoder 2) | $20.41_{\pm 1.837}$ | $24.73_{\pm 0.172}$ | $0.21_{\pm 0.078}$ | $0.2_{\pm 0.077}$ | $0.2_{\pm 0.079}$ |
| | iii (classifier encoding) | $24.38_{\pm 1.71}$ | $24.71_{\pm 0.15}$ | $0.44_{\pm 0.117}$ | $0.41_{\pm 0.134}$ | $0.42_{\pm 0.124}$ |
| | iv (no encoding) | $12.42_{\pm 1.25}$ | $24.73_{\pm 0.149}$ | $0.08_{\pm 0.073}$ | $0.11_{\pm 0.075}$ | $0.09_{\pm 0.081}$ |
| Fashion-MNIST | i (autoencoder 1) | $25.22_{\pm 1.365}$ | $14.92_{\pm 0.048}$ | $0.53_{\pm 0.065}$ | $0.53_{\pm 0.065}$ | $0.51_{\pm 0.06}$ |
| | ii (autoencoder 2) | $25.84_{\pm 1.436}$ | $14.85_{\pm 0.03}$ | $0.57_{\pm 0.059}$ | $0.58_{\pm 0.057}$ | $0.56_{\pm 0.055}$ |
| | iii (classifier encoding) | $27.23_{\pm 1.44}$ | $14.84_{\pm 0.037}$ | $0.64_{\pm 0.052}$ | $0.62_{\pm 0.056}$ | $0.62_{\pm 0.049}$ |
| | iv (no encoding) | $20.83_{\pm 1.317}$ | $14.95_{\pm 0.043}$ | $0.42_{\pm 0.14}$ | $0.44_{\pm 0.15}$ | $0.41_{\pm 0.132}$ |

- *Accuracy:* the biggest drop is for “no encoding” (which are the most easily recognizable
- *Transferability*: the accuracy is (essentially) the same for all CNN

UNIVERSITÄT LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Results – Quantitative (takeaway)

TABLE I. Results for MNIST and FashionMNIST.

| Dataset | Generation Method | $\|\|\mathcal{A}-\mathcal{T}\|\|$ Similarity to $\mathcal{T}$ | $\|\|\mathcal{A}-\mathcal{O}\|\|$ Similarity to $\mathcal{O}$ | $Acc$(CNN) VGG-11 | $Acc$(CNN) VGG-13 | $Acc$(CNN) Resnet-10 |
|---|---|---|---|---|---|---|
| MNIST | i (autoencoder 1) | $19.87_{\pm 1.794}$ | $24.85_{\pm 0.11}$ | $0.28_{\pm 0.081}$ | $0.26_{\pm 0.079}$ | $0.27_{\pm 0.084}$ |
| | ii (autoencoder 2) | $20.41_{\pm 1.837}$ | $24.73_{\pm 0.172}$ | $0.21_{\pm 0.078}$ | $0.2_{\pm 0.077}$ | $0.2_{\pm 0.079}$ |
| | iii (classifier encoding) | $24.38_{\pm 1.71}$ | $24.71_{\pm 0.15}$ | $0.44_{\pm 0.117}$ | $0.41_{\pm 0.134}$ | $0.42_{\pm 0.124}$ |
| | iv (no encoding) | $12.42_{\pm 1.25}$ | $24.73_{\pm 0.149}$ | $0.08_{\pm 0.073}$ | $0.11_{\pm 0.075}$ | $0.09_{\pm 0.081}$ |
| Fashion-MNIST | i (autoencoder 1) | $25.22_{\pm 1.365}$ | $14.92_{\pm 0.048}$ | $0.53_{\pm 0.065}$ | $0.53_{\pm 0.065}$ | $0.51_{\pm 0.06}$ |
| | ii (autoencoder 2) | $25.84_{\pm 1.436}$ | $14.85_{\pm 0.03}$ | $0.57_{\pm 0.059}$ | $0.58_{\pm 0.057}$ | $0.56_{\pm 0.055}$ |
| | iii (classifier encoding) | $27.23_{\pm 1.44}$ | $14.84_{\pm 0.037}$ | $0.64_{\pm 0.052}$ | $0.62_{\pm 0.056}$ | $0.62_{\pm 0.049}$ |
| | iv (no encoding) | $20.83_{\pm 1.317}$ | $14.95_{\pm 0.043}$ | $0.42_{\pm 0.14}$ | $0.44_{\pm 0.15}$ | $0.41_{\pm 0.132}$ |

- *Accuracy:* the biggest drop is for "no encoding" (which are the most easily recognizable
- *Transferability*: the accuracy is (essentially) the same for all CNN
- *Similarity to* $\mathcal{T}$: *classifier encoding* are the least similar to $\mathcal{T}$

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Results – Quantitative (takeaway)

TABLE I. Results for MNIST and FashionMNIST.

| Dataset | Generation Method | $\|\|\mathcal{A}-\mathcal{T}\|\|$ Similarity to $\mathcal{T}$ | $\|\|\mathcal{A}-\mathcal{O}\|\|$ Similarity to $\mathcal{O}$ | $Acc$(CNN) VGG-11 | $Acc$(CNN) VGG-13 | $Acc$(CNN) Resnet-10 |
|---|---|---|---|---|---|---|
| MNIST | i (autoencoder 1) | $19.87_{\pm 1.794}$ | $24.85_{\pm 0.11}$ | $0.28_{\pm 0.081}$ | $0.26_{\pm 0.079}$ | $0.27_{\pm 0.084}$ |
| | ii (autoencoder 2) | $20.41_{\pm 1.837}$ | $24.73_{\pm 0.172}$ | $0.21_{\pm 0.078}$ | $0.2_{\pm 0.077}$ | $0.2_{\pm 0.079}$ |
| | iii (classifier encoding) | $24.38_{\pm 1.71}$ | $24.71_{\pm 0.15}$ | $0.44_{\pm 0.117}$ | $0.41_{\pm 0.134}$ | $0.42_{\pm 0.124}$ |
| | iv (no encoding) | $12.42_{\pm 1.25}$ | $24.73_{\pm 0.149}$ | $0.08_{\pm 0.073}$ | $0.11_{\pm 0.075}$ | $0.09_{\pm 0.081}$ |
| Fashion-MNIST | i (autoencoder 1) | $25.22_{\pm 1.365}$ | $14.92_{\pm 0.048}$ | $0.53_{\pm 0.065}$ | $0.53_{\pm 0.065}$ | $0.51_{\pm 0.06}$ |
| | ii (autoencoder 2) | $25.84_{\pm 1.436}$ | $14.85_{\pm 0.03}$ | $0.57_{\pm 0.059}$ | $0.58_{\pm 0.057}$ | $0.56_{\pm 0.055}$ |
| | iii (classifier encoding) | $27.23_{\pm 1.44}$ | $14.84_{\pm 0.037}$ | $0.64_{\pm 0.052}$ | $0.62_{\pm 0.056}$ | $0.62_{\pm 0.049}$ |
| | iv (no encoding) | $20.83_{\pm 1.317}$ | $14.95_{\pm 0.043}$ | $0.42_{\pm 0.14}$ | $0.44_{\pm 0.15}$ | $0.41_{\pm 0.132}$ |

- *Accuracy:* the biggest drop is for “no encoding” (which are the most easily recognizable
- *Transferability*: the accuracy is (essentially) the same for all CNN
- *Similarity to $\mathcal{T}$*: *classifier encoding* are the least similar to $\mathcal{T}$
- *Similarity to $\mathcal{O}$*: all methods appear to have same results

UNIVERSITÄT LIECHTENSTEIN

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

# Future Work

- **Human evaluation**
  - We want to submit the adversarial samples $\mathcal{A}$ to real humans and ask for their opinion
- **Defense** and **augmentation**
  - Through *adversarial training*, it is possible to use $\mathcal{A}$ to defend against similar attacks
  - Alternatively, it is possible to use $\mathcal{A}$ to augment the training dataset and (potentially) increase the baseline performance of the CNN
- **Different data**
  - We only considered MNIST and FashionMNIST, but more datasets exist (e.g., CIFAR) which can be used to devise more intriguing experiments (with real FIAT Pandas and VW Polos!)
- **Other domains**
  - We only investigated CNN that were analyzing images. However, the same principles can be applied also in other domains (i.e., malware analysis)

IEEE Symposium on Security and Privacy

Deep Learning and Security Workshop (May 26th, 2022)

# Concept-Based Adversarial Attacks: Tricking Humans and Classifiers Alike

Johannes Schneider, Giovanni Apruzzese

UNIVERSITÄT LIECHTENSTEIN