

# Practice Lab 2

## General Rules

If you decide to split the workload within the group, make sure to explain to each other what you've done. Each group member should be able to present the whole report.

## Code

- Don't use AI-generated code unless you can explain exactly what it does
- Remove any experimental / commented-out code from your submission

## Report structure

1. Methodology: Implementation details and experimental design
2. Results: Quantitative and qualitative findings with visualisations
3. Analysis: Answer all questions posed in the tasks

## Getting help

- Use the relevant channels on the course Discord server for discussing the assignment
- When getting advice from other groups, don't simply copy their code
- Only send direct messages to teachers regarding personal matters (no general questions!)

## PART I: Model Inversion with RMIA

Robust Membership Inference Attack (RMIA) tries to determine whether any given sample was part of the training data of the target model. You will make your own implementation of this attack against an instance of ResNet-18 trained on the CIFAR-10 dataset. Both the model and dataset are easily available through `torchvision` and should be used with `PyTorch`.

### Task 1.1: Implementing the attack

You will need a collection of data samples that were used for training the target model as well as samples that were not used. You should split off and reserve a part of the dataset before training the model.

You do not need to implement an online attack mode, offline is sufficient.

HINT: The RMIA [paper](#) provides a clear conceptual procedure for the attack implementation.

You may look at existing implementations for inspiration but do not copy their code (wholly or partially) into your submission. You must write your own implementation!

### Task 1.2: Analysis and Evaluation

Design and run experiments to answer these questions:

1. How close do your results get to the paper? Evaluate your attack in terms of FPR vs TPR rate as well as AUROC for comparison.

2. How does the number of reference models affect the attack's success? Is there an ideal number?
3. What happens if you deliberately create class imbalance when setting aside data before training?

## **PART II: Obfuscation with HRR**

### **Task 2.1: Defending against the attack**

Implement and test the 'Holographically Reduced Representations' (HRR) defense mechanism from [this paper](#) and test it against your attack implementation from part 1. Your version can simplify the paper's pipeline by:

1. Replacing the central Unet with the ResNet-18 model you used before.
2. Leaving out the adversarial network that tries to predict the output class without the secret  $s$ .
3. Ignoring additional measures to mitigate accuracy loss.

HINT: You might need to adjust ResNet-18's architecture to smoothly replace the Unet.

### **Task 2.2: Analysis and Evaluation**

Questions to investigate:

1. How effective is HRR at preventing RMIA from succeeding?
2. Does HRR qualify as encryption? (Motivate your opinion)
3. Could an attacker adapt their strategy to overcome this defense?

**Congrats, you are now done with Lab 2!**

**You have just run a membership inference attack against a decent model and learned how to protect the model by obscuring the data during training.**

**Well done!**