
Deep Learning

Project proposals

Choose the projects [here](#)

Reading group on generalization

Context:

1. Research team in Bitdefender on this topic: <https://bit-ml.github.io/>
2. CoLLAs - **Conference on Lifelong Learning Agents** - Small conference, but very targeted, leading to meaningful/productive interactions
 - Next year most likely it will be organized in Romania <https://lifelong-ml.cc/> (Razvan Pascanu is in the board)

Reading group:

- 0.5-1 paper/week
 - In person @UB
 - [Fill this form](#) until tomorrow 23:59 if you commit to participate in **at least in 4 out of the first 5 meetings**.
 - If there is enough interest, we will write back to you to decide together the details!
-

Out-of-Distribution Detection

Why it matters:

- Avoid prediction on unfamiliar data: identifying unknown inputs can prevent dangerous decisions (e.g. healthcare, autonomous driving)











References: [OpenOOD](#) benchmark, [FS-OOD](#) benchmark

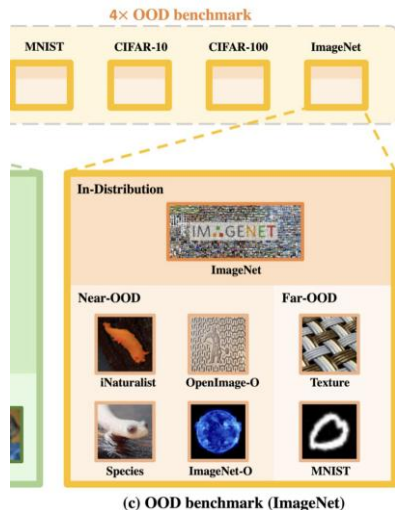
Datasets:

- ID: ImageNetV2 or Tiny ImageNet
- Near-OOD, Far-OOD

Methods: [Energy OOD](#), [DICE](#), [SCALE](#), [FDBD](#) (various, based on: distance, density, classif.)

- Compare several existing methods
- (optional, HARD) Improve over them

	Train			Val (OOD)	Test (OOD)
	d = Hospital 1	d = Hospital 2	d = Hospital 3	d = Hospital 4	d = Hospital 5
y = Normal					
y = Tumor					



Fingerprinting LLM Training Data

Text X

the 15th Miss
Universe Thailand
pageant was held at
Roval Paroon Hall



GPT-3.5

is pretrained on X

Task: detect if a LLM was trained on my document

Dataset: [MIMIR](#)

LLM Models: [Pythia](#) (16 LLMs; public data; 70M-12B params)

Models: e.g. Loss, zlib, Min-K% Prob, MIA-based methods

- Compare several existing methods
- (optional, SF) Improve over them

References:

<https://arxiv.org/abs/2310.15007>

<https://swj0419.github.io/detect-pretrain.github.io/>

<https://arxiv.org/abs/2406.06443>

<https://arxiv.org/abs/2406.17975>

<https://openreview.net/forum?id=PAWQvrForJ>

<https://openreview.net/forum?id=av0D19pSkU#discussion>

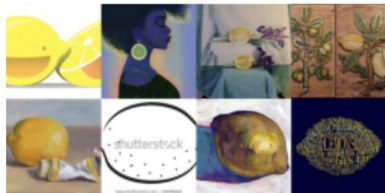
ImageNet (Deng et al.)



ImageNet Sketch (Wang et al.)



ImageNet-R (Hendrycks et al.)



Imagenet Fine-tuning and OOD Generalization




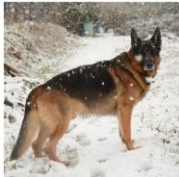
Description: Start with pretrained models (e.g. from torchvision or huggingface). Prevent the model from losing its generalization capabilities.

Setup: Train on: ImageNet(validation / V2). Test on: ImageNet-A, ImageNet-R, ImageNet-S.

Tasks:

1. Compare several existing methods: (Medium) [WiSE-FT](#), (Medium) [LP-FT](#), (HARD) [SAFT](#)
2. (optional, HARD) Try to (marginally) improve existing approaches

Spurious Correlations Identification

	Husky	German Sheperd
Train		
Test		

Task: Reproduce [WASP](#) and apply it for text classification datasets

Text datasets: Offensive social media posts, Fake News, etc.

Foundation Model: [GTE](#), or any other models for text retrieval

Steps:

- Identify possible SCs learned from the chosen dataset(s)
- Validate that classifiers trained with ERM (LP & fine-tuned encoder) are affected by the identified SCs
- Retrain the classifiers and make them more robust to the identified SCs (e.g. using [GDRO](#), Resampling or [other](#) methods)
- (optional) Use/propose other scoring methods, make ablations
- (optional) Use other methods for SCI (e.g. [Lg](#), [B2T](#))

Grading:

- 70% for analyzing one dataset
- 100% for two or more datasets

Detection

Real/Fake?



Deep Fake Image

- **Task**
 - Test different backbone architectures for deep fake detection of face images. Check how the models trained to detect a particular type of fakes transfer to other generation methods. (e.g. models trained on gan-generated images transfer to diffusion-generated images).
 - Adapt [DeCLIP](#) model for image detection and compare results with other SOTA deepfake image detection models
- **Datasets**
 - [CelebAHQ](#)
 - [StyleGan2](#) sampled images
 - [LDM](#) sampled images
 - [P2](#) sampled images
- **Links**
 - [CNN-generated images are surprisingly easy to spot... for now](#)
 - [Are GAN generated images easy to detect? A critical analysis of the state-of-the-art.](#)
 - [What makes fake images detectable? Understanding properties that generalize](#)
 - [Towards the detection of diffusion model deepfakes](#)

Deep Fake Localization



- **Task**
 - Localize manipulated regions in images. You will experiment with different architectures for manipulation segmentation and analyze how results transfer from one deep fake generation method to another.
- **Datasets**
 - [Repaint](#)
 - [LAMA](#)
 - [Pluralistic](#)
- **Links**
 - [CNN-generated images are surprisingly easy to spot... for now](#)
 - [Are GAN generated images easy to detect? A critical analysis of the state-of-the-art.](#)
 - [What makes fake images detectable? Understanding properties that generalize](#)
 - [Towards the detection of diffusion model deepfakes](#)

Detection

Real/Fake?



General Deepfake

- **Context.** One of the main challenges in deepfake detection is building a detector capable of generalizing to images produced with different generation methods than those seen at training.
- **Task.** Investigate the generalization capabilities of pretrained self-supervised features for deepfake image detection.
- **Details.** You will experiment with and extend the method presented in this [paper](#). You can start from this [code](#) that provides a trained linear classification layer on top of features extracted from [CLIP](#). This model is trained on GAN-generated data. Your tasks will be to:
 - Test the provided model for classification of deepfake images from latest diffusion models. Specifically, you will use for testing the validation test provided in this [challenge](#). Compare the obtained results with those provided on other diffusion models.
 - Modify the approach to rely on image features extracted from other powerful image encoders (eg. [SAM](#)) instead of CLIP
 - a) Train a classifier on top of these features using GAN-data.
 - b) Compare the results obtained with CLIP and those obtained with SAM features.

Audio-Video Deepfake Detection using pre trained feature extractor

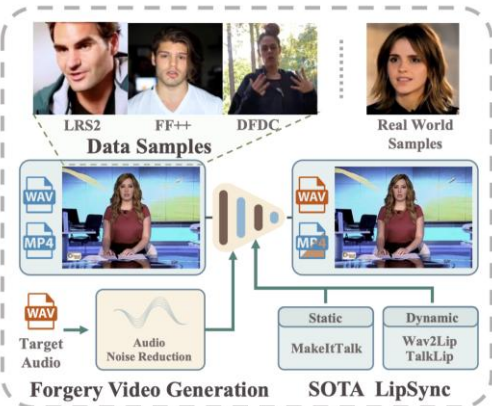
- **Task:** use a model pre trained on an audio-video task (e.g. [AV-HuBERT](#)) as a feature extractor. Using these features, predict either a video is real or fake.
- **Expectation:**
 - Extract audio-only and video-only features using pre trained feature extractor.
 - As a baseline, run a simple cosine similarity between the two (dissimilarities may indicate deepfakes).
 - Use features from different layers to train simple neural networks to predict if a video is real or fake.

Resources:

- [AV-HuBERT](#)
- [AVLips](#) dataset

Scoring:

- 80%
 - Correctly train a simple model in the given setup
 - Relevant plots for monitoring train/test/validation performance
- 100%
 - Propose ways to improve performance
 - Thorough analysis between multiple approaches (both in terms of model trained and features used)



Cross-domain generalization of modern text encoders

- **Research q:** Do modern BERT-style encoders perform better on cross-domain tasks?
- **Task:** Evaluate models in cross-domain setups (e.g train on general news, test on financial news)
- **Expectation:**
 - evaluate how ModernBERT performs on cross-domain B when trained on domain A
 - evaluate if ModernBERT adapts better to cross-domain B when also adding data from B
 - full experimental setup [link](#)

Resources:

- [ModernBERT](#)

Scoring:

- 80%
 - train BERT&ModernBERT on source in- domain A and evaluate on both in-domain and cross domain B
 - plot confusion matrices for both models for both test sets (in domain and cross-domain)
- 100%
 - analysis for both models when further incorporating 10/20/50% cross-domain train data

Are modern text encoders more robust to noise?

- **Research q:** Are modern BERT-style encoders more robust to input noise?
- **Task:** train models on both original data and original+perturbed data (replaced synonyms, typos, word deletions)
- **Expectation:**
 - evaluate if ModernBERT is more robust to perturbations when trained on clean vs noised data
 - for a particular noise, investigate the robustness of model for different levels of corruption (2/5/10% noisy examples)
 - full experimental [setup](#)

Resources:

- [ModernBERT](#)

Scoring:

- 80%
 - train BERT and ModernBERT on several types of perturbations using both clean only data and noise-aware training
 - plot confusion matrices for all types of training
- 100%
 - analysis on different levels of noise
 - error analysis

Which Ro LLMs are the best feature extractors?

- **Research Q:** what Romanian LLMs are the best feature extractors on Romanian NLP tasks?
- **Task:** compare several RoLLMs vs general LLMs (LLama3) as feature extractors and train simple MLP classifiers on top of embeddings
- **Expectations:**
 - evaluate which RoLLM features perform best on downstream tasks?
 - evaluate which pooling strategy is the best? (avg token, last token, echo embedding etc.)
 - [Experimental setup](#)

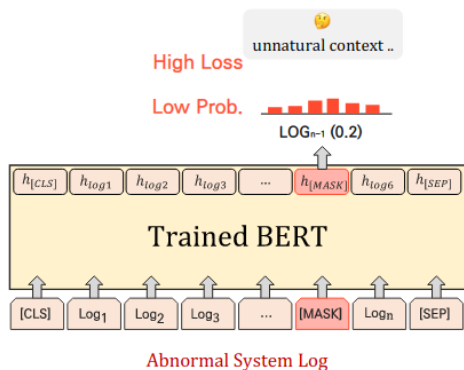
Resources:

- [echo embeddings](#), [summarisation technique](#)
- models: [mGPT-RO](#), [RoLLama2](#), [LLMic](#), Llama3

Scoring:

- **80%**
 - analyse embedding performance for LaRoSeDA and roARC challenge
- **100%**
 - analyse best pooling strategy
 - plot embeddings for best model for all 4 pooling techniques

Anomaly detection in network logs



Anomaly score computation (Source: LAnoBERT)

- **Task:** Discriminate anomalous logs (outliers) from benign logs (inliers) using language modelling in unsupervised or self-supervised fashion

- **Expectation:**

- Use a **RNN (LSTM/GRU)** or a **Transformer Model (BERT)** to learn the inlier distribution
- Compute an **anomaly metric** for the test samples and evaluate the model using **ROC-AUC** between your anomaly scores and dataset labels

- **Dataset:**

- **CICIDS2018:**
<https://www.unb.ca/cic/datasets/ids-2018.html>

Resources:

- RNN for log anomaly detection: <https://arxiv.org/pdf/1803.04967.pdf>
- LAnoBERT: <https://arxiv.org/pdf/2111.09564.pdf>
- OE: <https://arxiv.org/pdf/1812.04606.pdf>

Scoring:

- **80%**
 - Correctly train a simple model in the given setup
 - Correct plots for train/test splits
- **100%**
 - Obtain good performance
 - Analyze anomaly detection rate based on outlier class (CICIDS2018 labels: Bruteforce attack, DoS attack, Botnet attack etc.)
 - Comparison: split data by timestamp (test on different period than the train data) vs random split
- **Bonus:** Outlier Exposure

Conspiracy detection PAN 2024

Murder: most coronavirus patients **died** due to **treatment** on ventilators... In a recent study, researchers **tracked** people **infected** with covid-19, and 320 patients required **hospitalization**. However, 88 percentage of them **died** while being **treated** on ventilators... It's time to stop the ventilator treatment!

conspiratorial text
critical text

- **Task:** given a text, detect if it's conspiracy or public messaging
- **Resources:**
<https://pan.webis.de/clef24/pan24-web/oppositional-thinking-analysis.html>
- **Grading guidelines**
 - 70% - finetune multilingual BERT
 - 100% - compare multilingual to language specific models (BERT/BETO)
 - perform feature analysis
 - experimental setup [link](#)

Detect human or machine text

Hey there, great to meet you. I'm Pi, your personal AI.

My goal is to be useful, friendly and fun. Ask me for advice, for answers, or let's talk about whatever's on your mind.

How's your day going?

machine generated
human

- **Task:** given a text, determine if it was generated by a model or written by a human (**2 classes**)
- **Resources**
 - [paper](#):
 - datasets: [PAN](#), [CodaLab](#)
 - experimental setup [link](#)
- **Grading guidelines**
 - 70% - finetune BERT and perform evaluate it cross-domain
 - 100% - perform feature analysis to understand domain-specific cues and address them

Steps for the Project

1. [Select a project](#), team of 2 (hard deadline for choosing project: **April 2**)
 - a. If you are not the first team for a project, ask the mentor if you can choose it
2. Follow the instructions from your **mentor**
 - a. Read the **Related Work**
 - b. Choose a **Dataset**
 - c. Implement several **Baselines** in Pytorch
 - d. **Comparative Analysis** - variations for:
 - i. Architecture
 - ii. Number of parameters
 - iii. Optimization, cost function
3. **Poster** for the **Project** (presented in the final week)

Bonus: Submit for eeml.eu (deadline: **March 31 - may be extended for 1 week**)

Mandatory: **Deep Learning** approach for the chosen problem

—

—

—

Images courtesy of...

- <https://20bn.com/datasets/jester>
 - <https://github.com/maximecb/gym-minigrid>
 - <https://www.elmundotech.com/2015/02/25/googles-deep-mind-creates-an-ai-system-to-beat-video-games-by-itself/>
 - <https://aws.amazon.com/blogs/machine-learning/detecting-fraud-in-heterogeneous-networks-using-amazon-sagemaker-and-deep-graph-library/>
 - <https://www.dynamicciso.com/dark-nexus-the-evolving-iot-botnet-targets-variety-of-devices-says-bitdefender-research/>
 - <https://studentwork.prattsi.org/infovis/visualization/amazon-product-co-purchasing-network-information-visualization/>
 - https://github.com/tkarras/progressive_growing_of_gans
-