Abstract: Self-Supervised Anomaly Detection in Audio Spectrograms

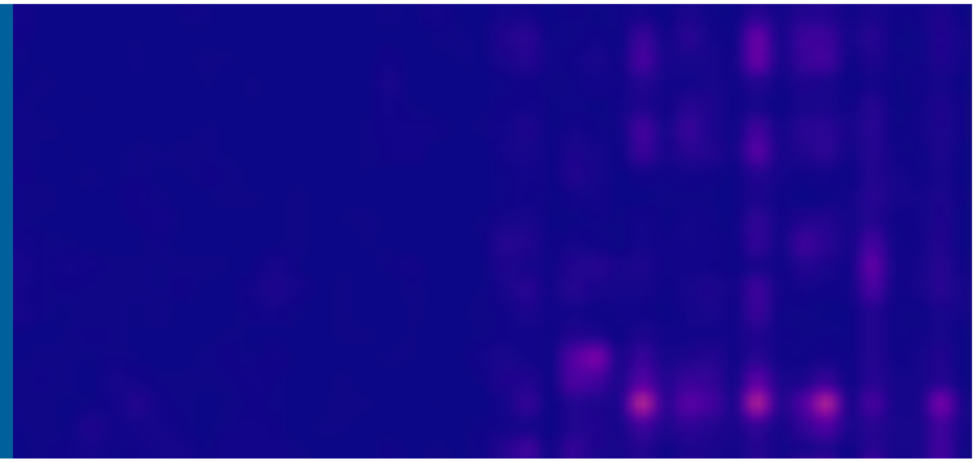
Samir Rajani

August 2022

Autoencoders are a powerful technique for anomaly detection whereby a neural network is trained to reconstruct its input from a low-dimensional latent representation. Samples with large reconstruction error and out-of-distribution embeddings at the bottleneck are likely candidates to be anomalies. In this project, we demonstrate the ability of a convolutional autoencoder with a U-net architecture to detect anomalies in forest recordings from the Morton Arboretum in Lisle, IL, both in the presence and in the absence of anomalies in training data. We also present briefly on joint embedding architectures, finding that monitoring the loss term in the variance-invariance-covariance regularization technique is not well-suited to the anomaly detection task. Finally, we present future directions for research, including the use of anomaly pruning to improve classifiers trained on data containing anomalies and the adaptation of vision transformers to an encoder-decoder architecture.

Argonne NATIONAL LABORATORY

SELF-SUPERVISED ANOMALY DETECTION IN AUDIO SPECTROGRAMS



SAMIR RAJANI
University of Chicago
SULI Intern, MCS Division
Mentors: Rajesh Sankaran, Dario Dematties

8/3/2022
Lemont, IL

INTRODUCTION

INTRODUCTION

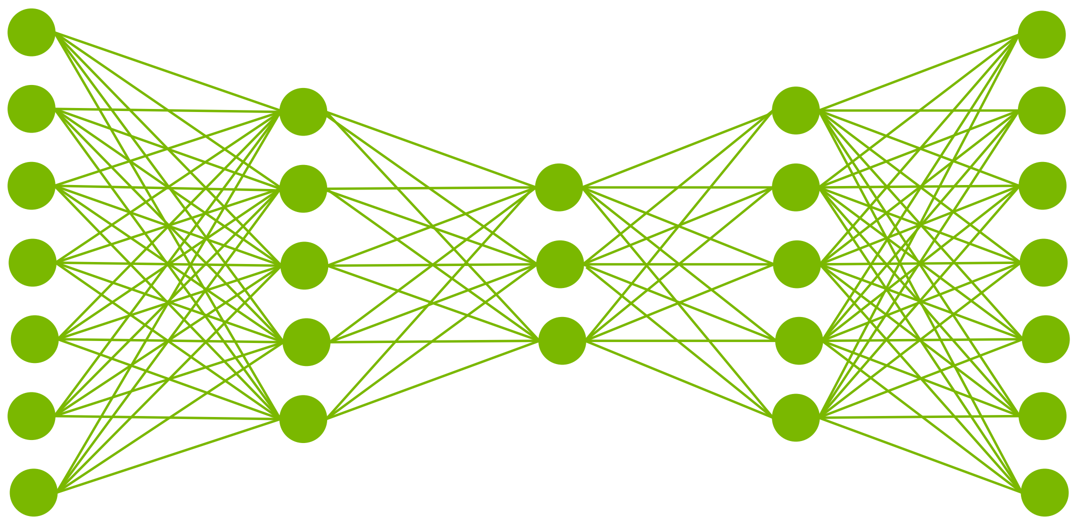
The Anomaly Detection Problem

- *Normal samples*: Classes of samples that occur frequently in a training dataset
- *Anomalous samples*: Classes of samples that occur rarely or never in a training dataset
- **Goal:** Detect anomalous samples in a test dataset
- **Issue:** When the dataset is not labelled, we cannot just train the neural network by telling it whether particular samples are “normal” or “anomalous”

INTRODUCTION

Autoencoders

- We can exploit the natural statistics of the data to build a robust anomaly detector!



Input Sample *Latent Representation* *Reconstructed Sample*

DATA AND PREPROCESSING

DATA AND PREPROCESSING

The BirdAudio Dataset

- Collected between 8/12/2021 and 8/28/2021
- Four six-hour .wav files per day
- Consist primarily of bird songs and noise from a nearby highway

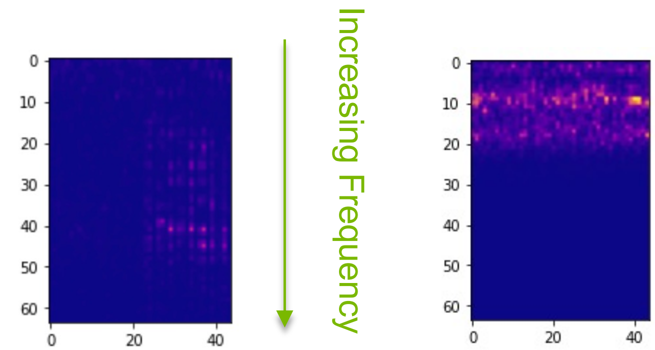
Waggle Project
Image from github.com/waggle-sensor

The Morton Arboretum, Lisle, IL
Image from mortonarb.org

DATA AND PREPROCESSING

Preprocessing and Transformations

- Resample to 22,050 Hz
- Mix down to a single channel
- Split into one-second clips and index clips
- Right-pad clips shorter than one second
- Apply Mel spectrogram transformation
- Optionally, crop frequencies of spectrogram:



A bird song, displayed in "tight crop" mode

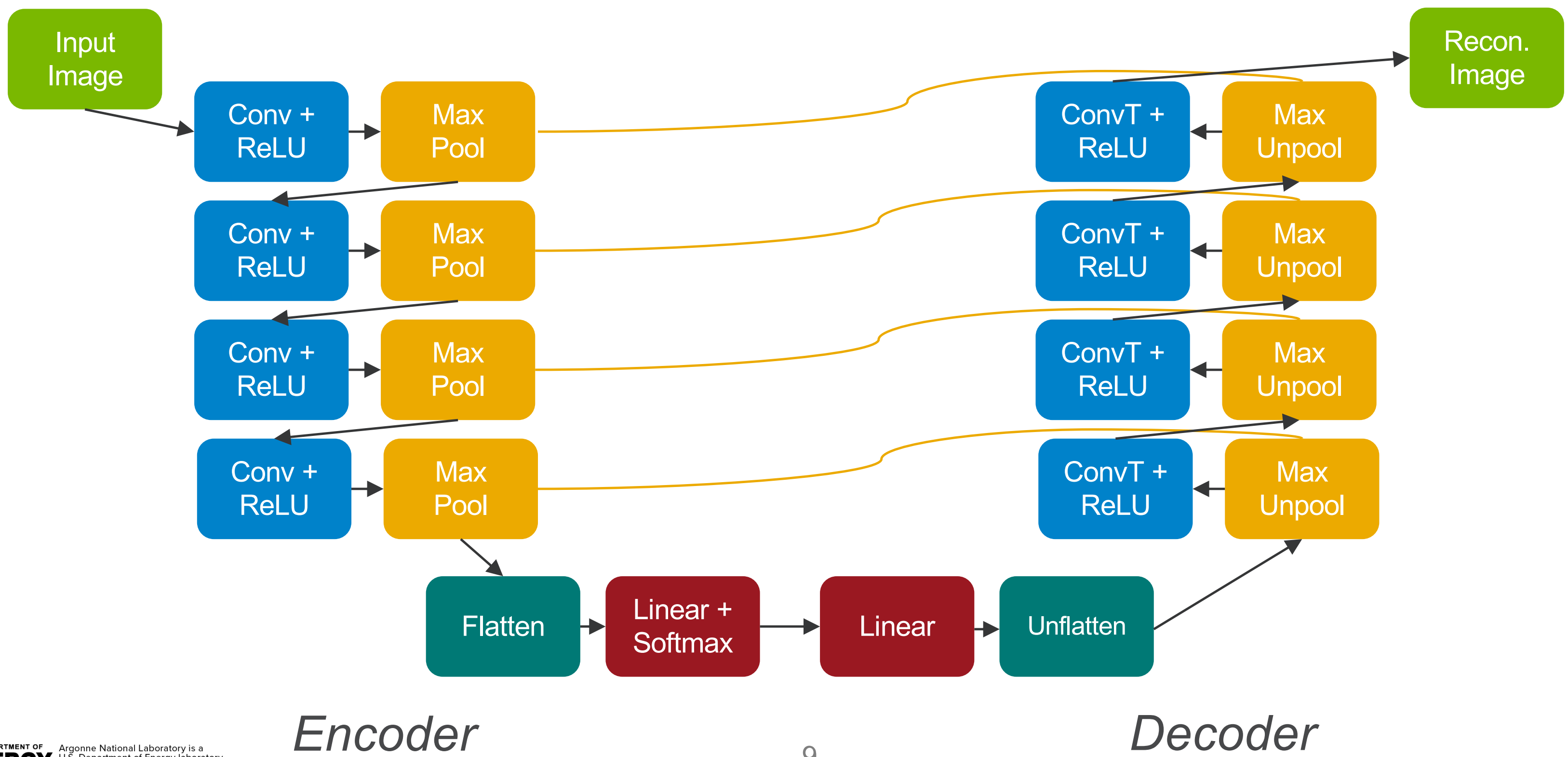
A spectrogram, displayed in "no crop" mode

Name	Purpose	# Mels	Mels Retained
No Crop	General-Purpose Anomaly Detection	64	0-63 (All)
Crop	Ignore Cars	128	64-127 (Second Half)
Tight Crop	Ignore Non-Birds	256	128-191 (Third Quarter)

NETWORK ARCHITECTURE

NETWORK ARCHITECTURE

A Custom U-Net



NETWORK ARCHITECTURE

A Custom U-Net: Parameters

Layer	Activation Shape	Activation Size	# Parameters
INPUT	(1, 64, 44)	2816	0
CONV1	(16, 66, 46)	48576	160
POOL1	(16, 33, 23)	12144	0
CONV2	(32, 35, 25)	28000	4640
POOL2	(32, 17, 12)	6528	0
CONV3	(64, 19, 14)	17024	18496
POOL3	(64, 9, 7)	4032	0
CONV4	(128, 11, 9)	12672	73856
POOL4	(128, 5, 4)	2560	0
FLATTEN	2560	2560	0

Layer	Activation Shape	Activation Size	# Parameters
FC1	10	10	25610
FC2	2560	2560	28160
UNFLATTEN	(128, 5, 4)	2560	0
UNPOOL1	(128, 11, 9)	12672	0
CONVT1	(64, 9, 7)	4032	73792
UNPOOL2	(64, 19, 14)	17024	0
CONVT2	(32, 17, 12)	6528	18464
UNPOOL3	(32, 35, 25)	28000	0
CONVT3	(16, 33, 23)	12144	4624
UNPOOL4	(16, 66, 46)	48576	0
CONVT4	(1, 64, 44)	2816	145

NETWORK ARCHITECTURE

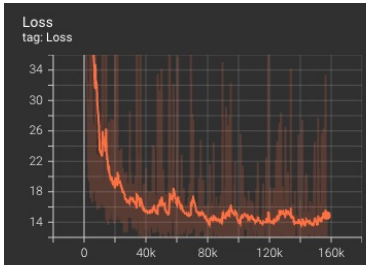
Hyperparameter Optimization

- Chose the best hyperparameters from the combinations of:
  - Learning Rate: 0.001, **0.0001**, 0.00001
  - Per-GPU Batch Size: 32, **256**
  - Weight Decay: 1e-5, 1e-6, **1e-7**
- Implemented distributed parallelization for training

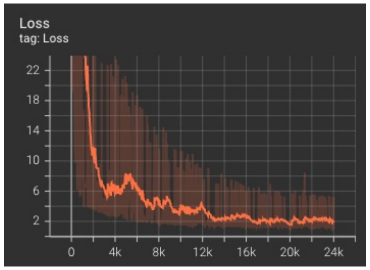
NETWORK ARCHITECTURE

Training

- Trained on 8 NVIDIA A100 GPUs on a single node at ALCF's ThetaGPU
- Trained separately on:
  - [2000 epochs] Fifteen audio files, not including the test audio file
  - [1000 epochs] Only the test audio file



Loss curve for 15-file training



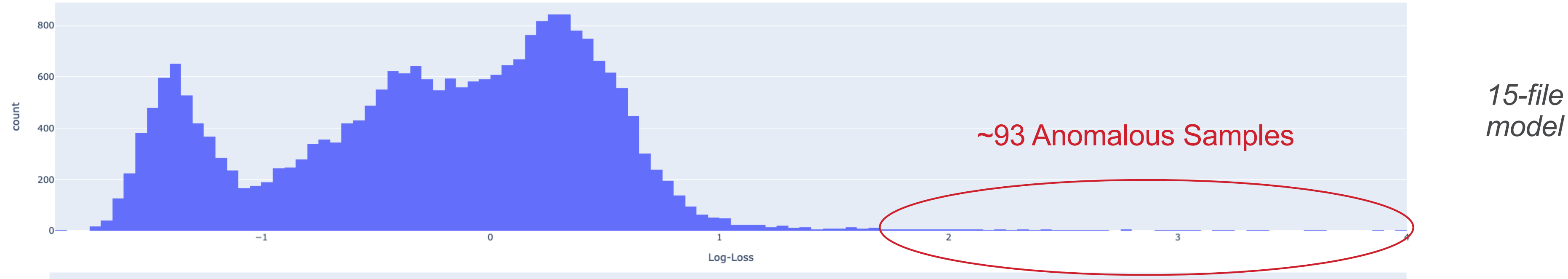
Loss curve for single-file training

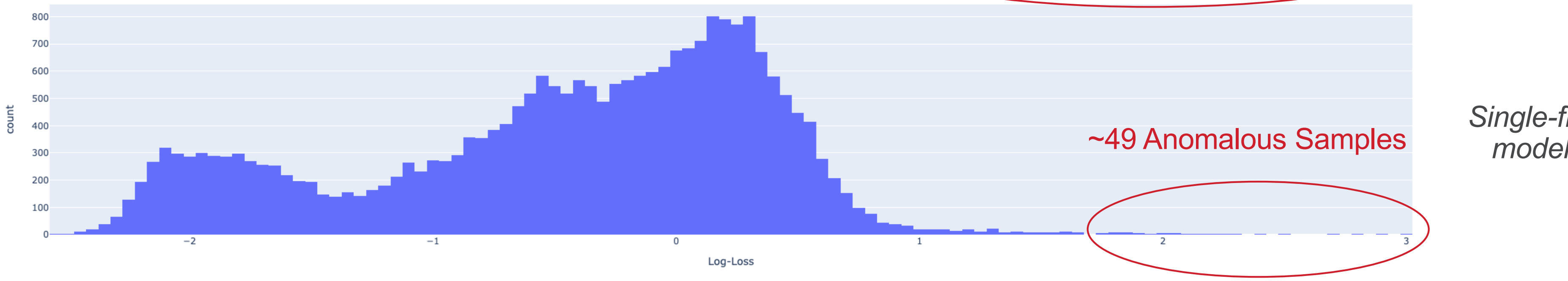
RESULTS

RESULTS

Analyzing Reconstruction Error

- Evaluated reconstruction error of ~24,000 one-second samples from test file

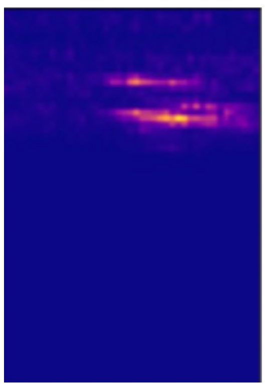




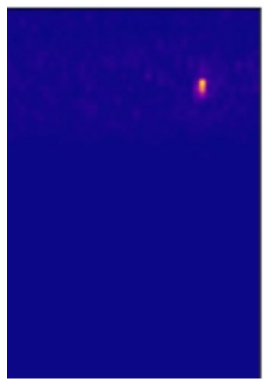
RESULTS

Analyzing Anomalous Samples

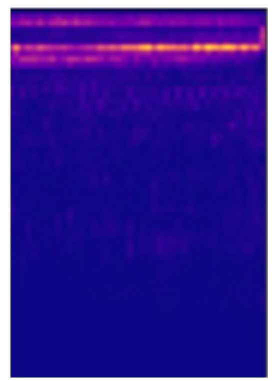
- What kinds of anomalies were found?



Motorcycle



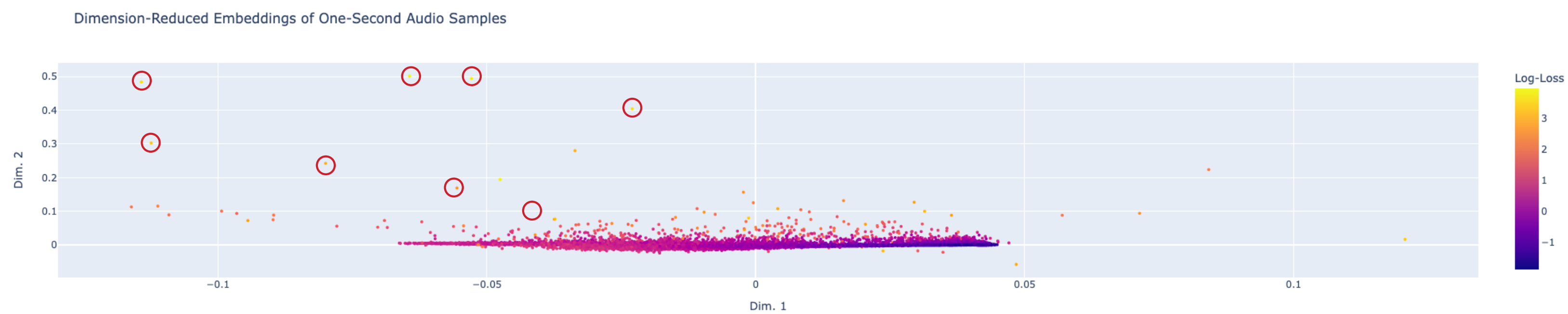
Percussive Noise



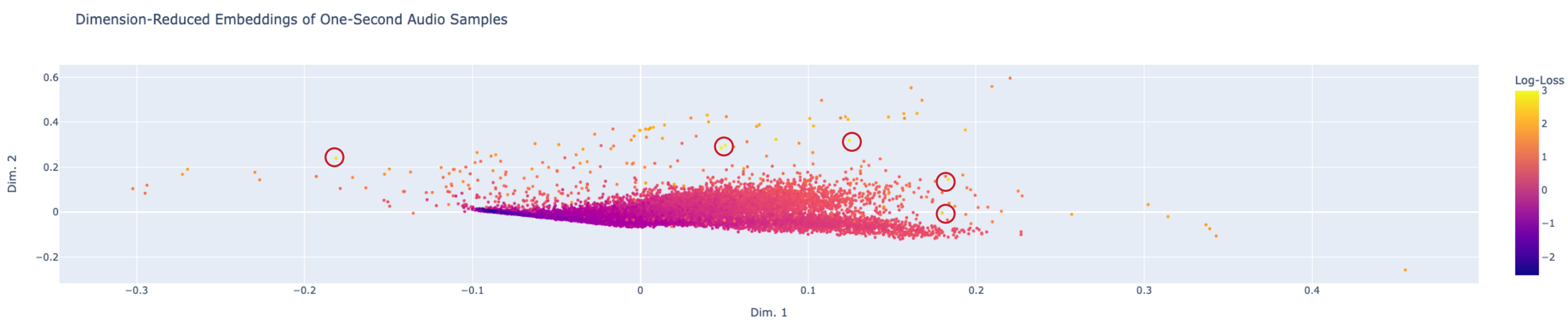
Plane Flying Overhead

RESULTS

Analyzing Embeddings



15-file model

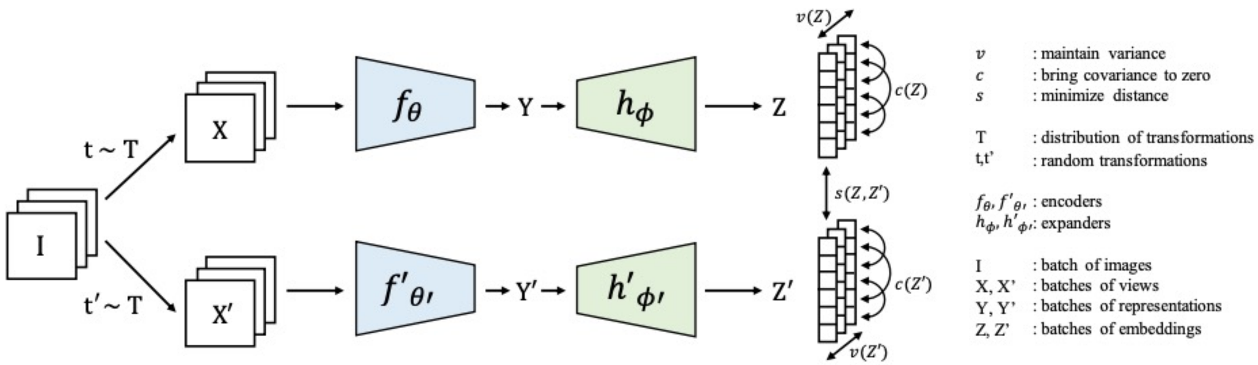


Single-file model

ASIDE: JOINT EMBEDDING

ASIDE: JOINT EMBEDDING

Joint Embedding Architectures for Anomaly Detection

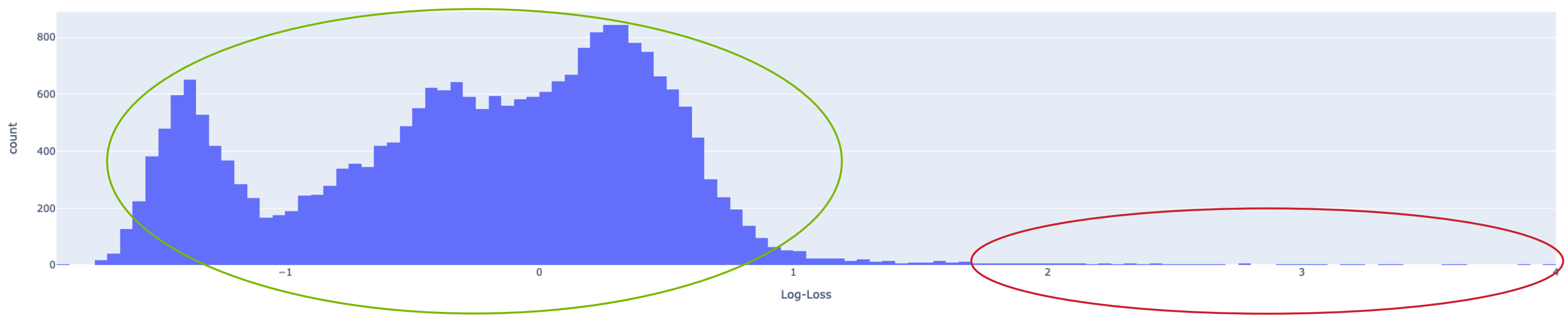


VICReg Architecture
Bardes et. al (2021)

FUTURE WORK

FUTURE WORK

ANOMALY PRUNING FOR ROBUST CLASSIFIERS



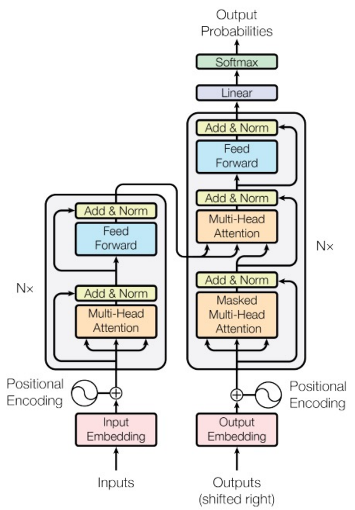
…potentially to the detriment of normal samples.

Anomalous samples provide much larger gradients…

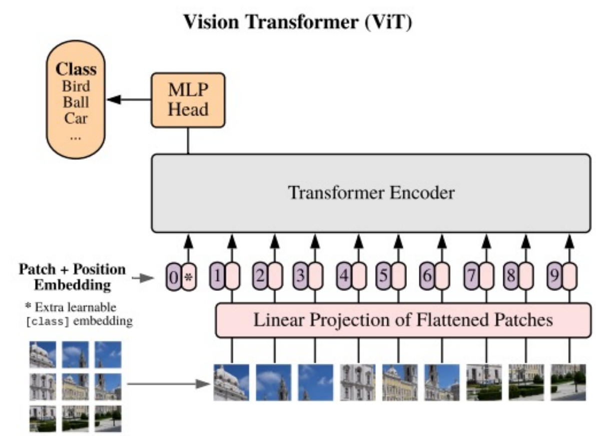
Remove samples with the highest reconstruction errors after every *n* epochs?

FUTURE WORK

Building an Autoencoding Transformer



Transformer Architecture
Vaswani et. al (2017)



Vision Transformer Architecture
Dosovitskiy et. al (2020)

THANK YOU FOR YOUR TIME!

Questions? Contact me at srajani@anl.gov.