

Extending Modality in Deep Autoencoding Gaussian Mixture Model for Anomaly Detection

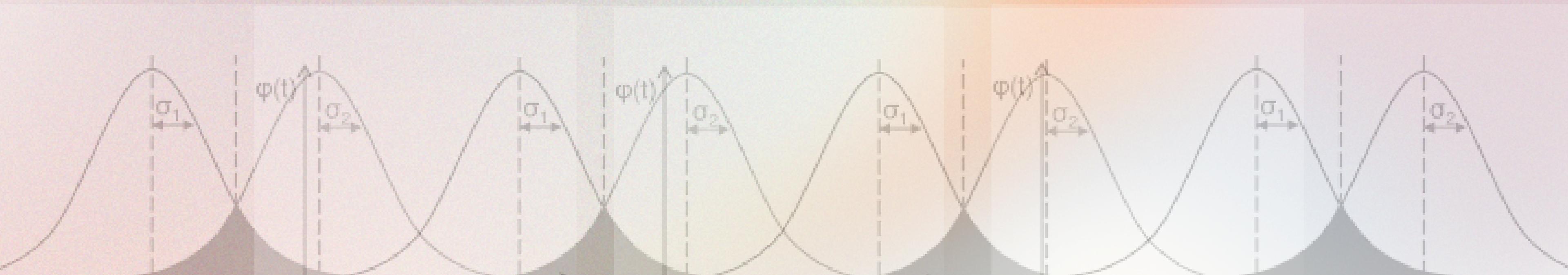
Aditya Jain | Aryan Solanki | Nupoor Assudani | Rishabh Jogani
23110016 | 23110049 | 23110224 | 23110276

CS 328 - Introduction to Data Science

Prof. Anirban Dasgupta, IIT Gandhinagar

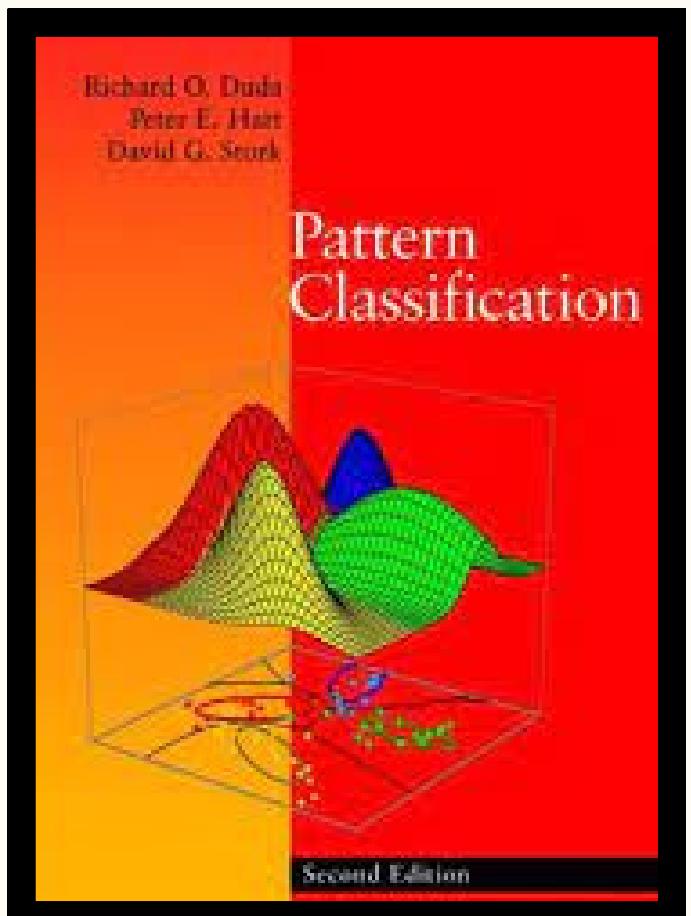
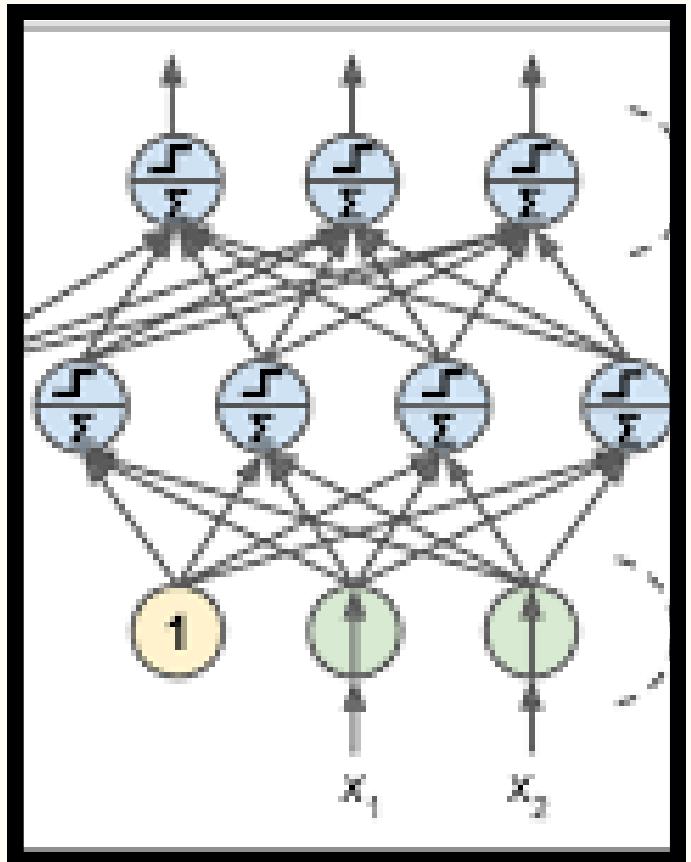
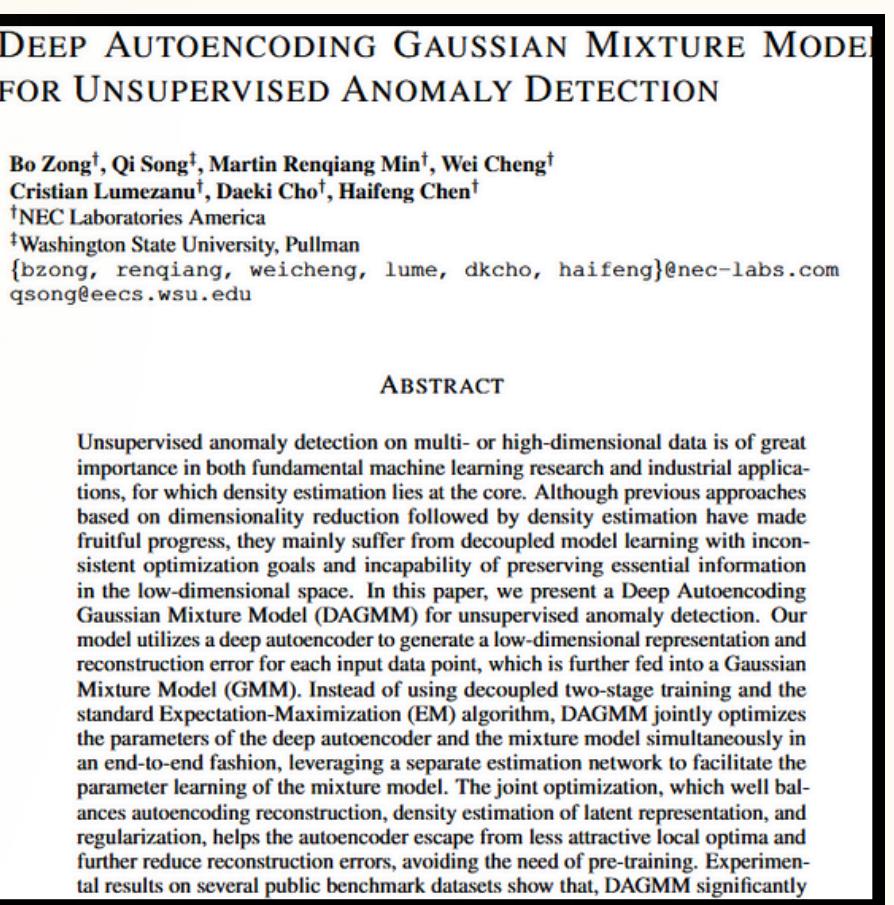
Overview

- Introduction
- What is a DAGMM? (ICLR 2018)
- Replication on benchmarks
- Current Research Gap
- Convolutional DAGMM (2 Frameworks)
- Evaluations
- The Audio Anomaly Challenge
- Future Prospects



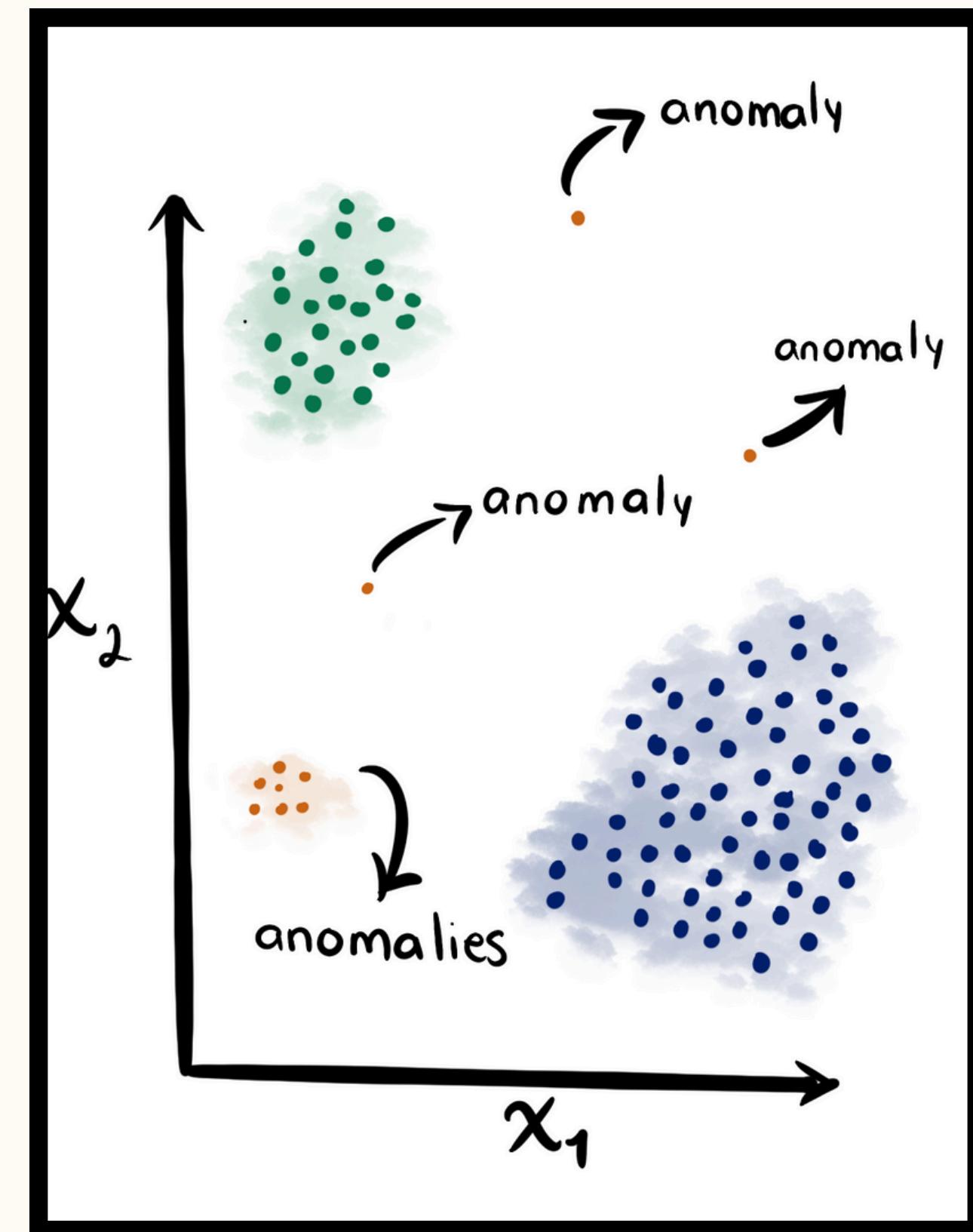
Introduction

- 1809 – Carl formalized the Gaussian distribution while modeling errors in astronomical data.
- 1973 – Duda & Hart introduced Gaussian Mixture Models (GMMs) in Pattern Classification and Scene Analysis.
- 1986 – Rumelhart, Hinton & Williams published the backpropagation method in Nature, forming the basis of MLPs.
- 2018 – ICLR : DAGMM combined autoencoders with GMMs for unsupervised anomaly detection.



Unsupervised Anomaly Detection

- Unsupervised anomaly detection identifies unusual patterns or outliers in data without using labelled examples of anomalies.
- The model learns the normal data distribution during training and flags deviations from this norm during inference.
- This approach is especially useful when anomalies are rare or hard to label.



GMMs are powerful

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Definitions

$$\pi_k \geq 0, \quad \sum_{k=1}^K \pi_k = 1$$

Mixing coefficient for the k -th Gaussian component.

$$\boldsymbol{\mu}_k \in \mathbb{R}^d$$

Mean vector of the k -th Gaussian.

$$\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}, \quad \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_k^\top, \quad \boldsymbol{\Sigma}_k \succ 0$$

Covariance matrix of the k -th Gaussian (symmetric and positive definite).

$$\mathbf{x} \in \mathbb{R}^d$$

Input data point.

$$\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right)$$

Probability density function of the multivariate Gaussian distribution.

EM Algorithm

The **Expectation-Maximization (EM)** algorithm alternates between two steps:

E-step

Compute the responsibilities $\gamma_k(\mathbf{x})$, which represent the probability that data point \mathbf{x} was generated by component k :

$$\gamma_k(\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

M-step

Maximize the log-likelihood with respect to the parameters π_k , $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$:

- Mixing coefficients π_k :

$$\pi_k = \frac{1}{N} \sum_{n=1}^N \gamma_k(\mathbf{x}_n)$$

where N is the total number of data points.

- Mean vectors $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma_k(\mathbf{x}_n) \mathbf{x}_n}{\sum_{n=1}^N \gamma_k(\mathbf{x}_n)}$$

This is a weighted average of the data points, where the weights are the responsibilities $\gamma_k(\mathbf{x}_n)$.

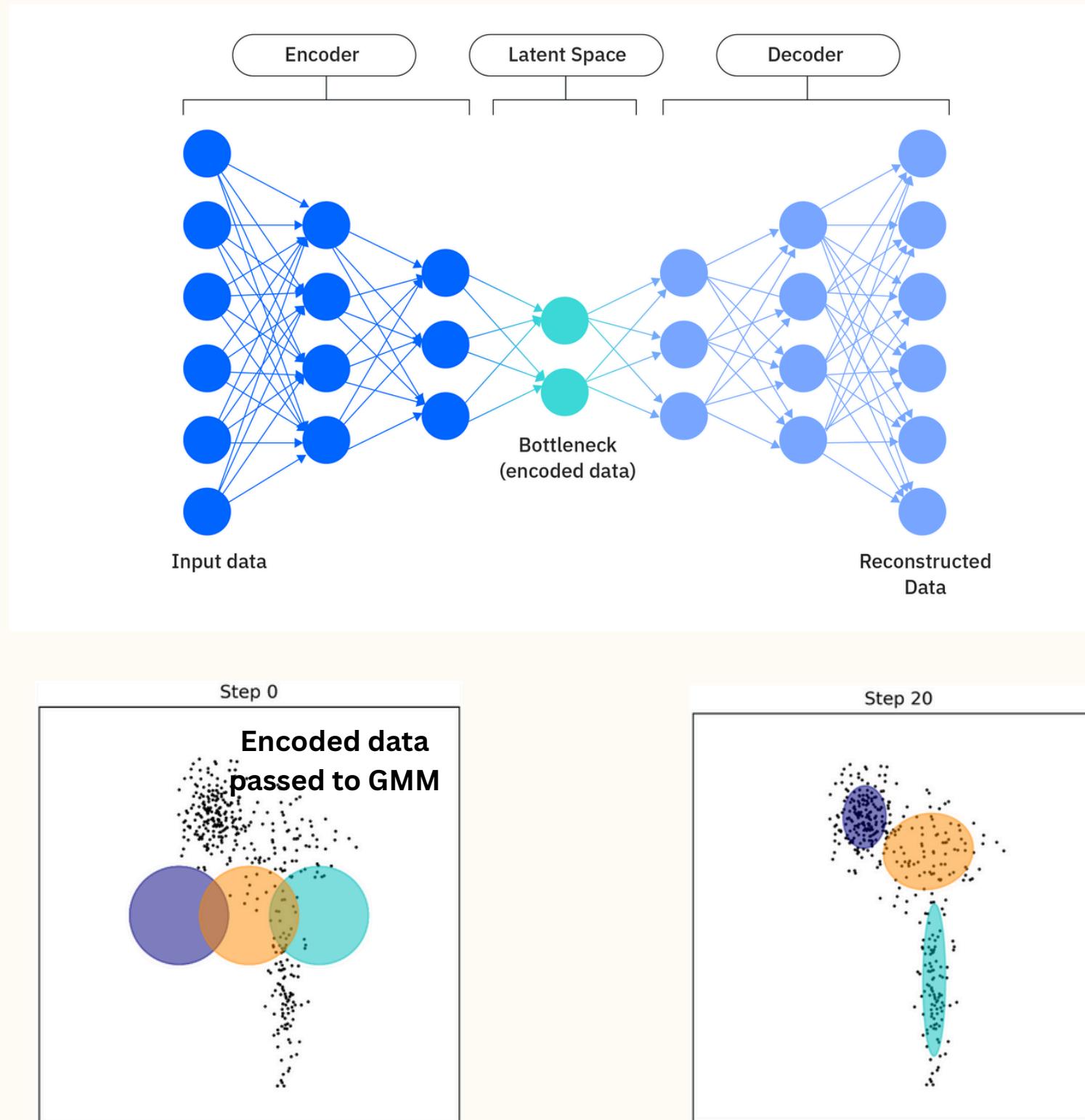
- Covariance matrices $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N \gamma_k(\mathbf{x}_n) (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top}{\sum_{n=1}^N \gamma_k(\mathbf{x}_n)}$$

This is the weighted covariance, where again the weights are the responsibilities $\gamma_k(\mathbf{x}_n)$.

- GMMs are powerful because they can model complex, multimodal data distributions by combining multiple Gaussian components. This allows them to capture clusters with varying shapes and densities. GMMs provide probabilistic outputs, making them ideal for tasks like anomaly detection where uncertainty is important.

Autoencoder + GMM



Multivariate Gaussian in High Dimension

Deep Autoencoding Gaussian Mixture Model

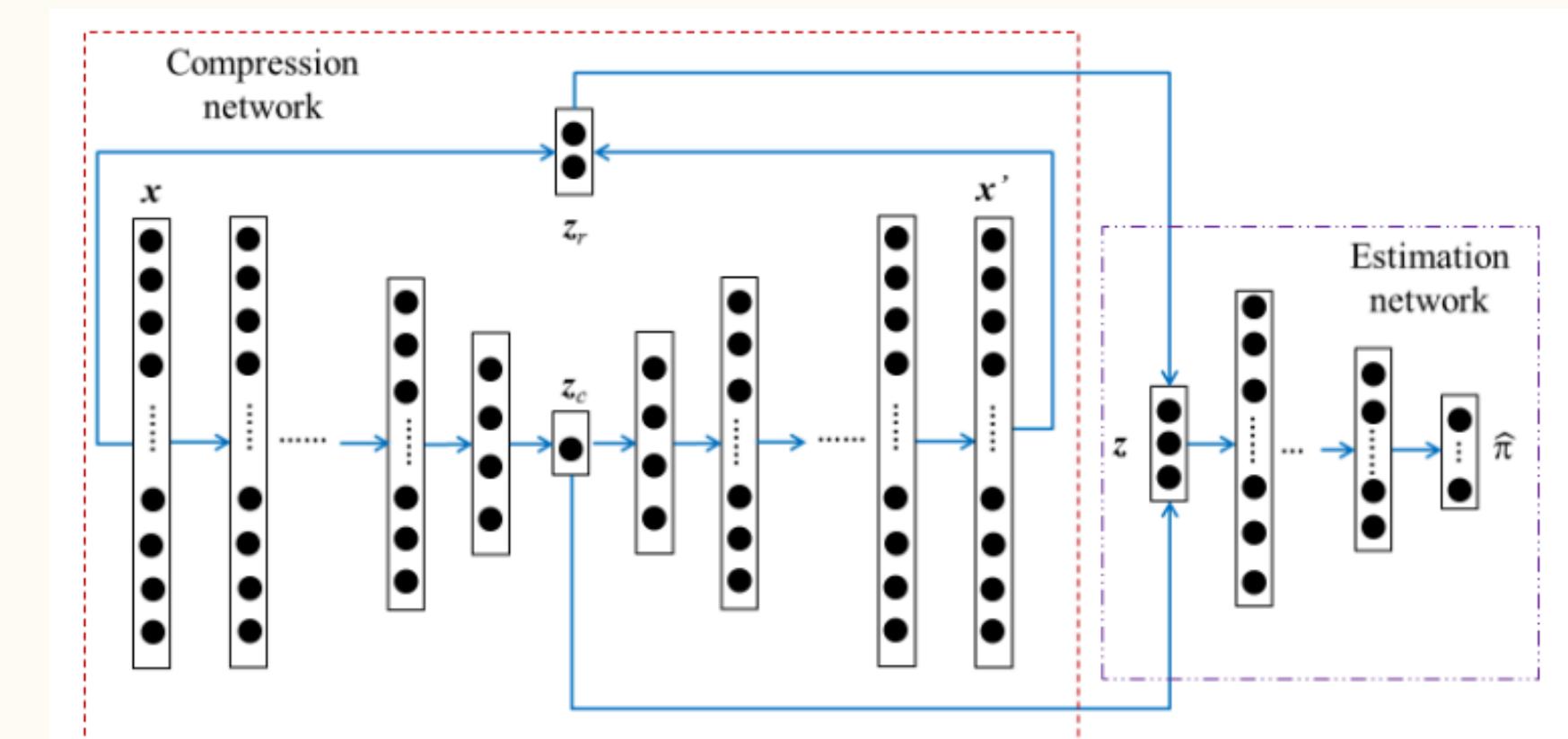
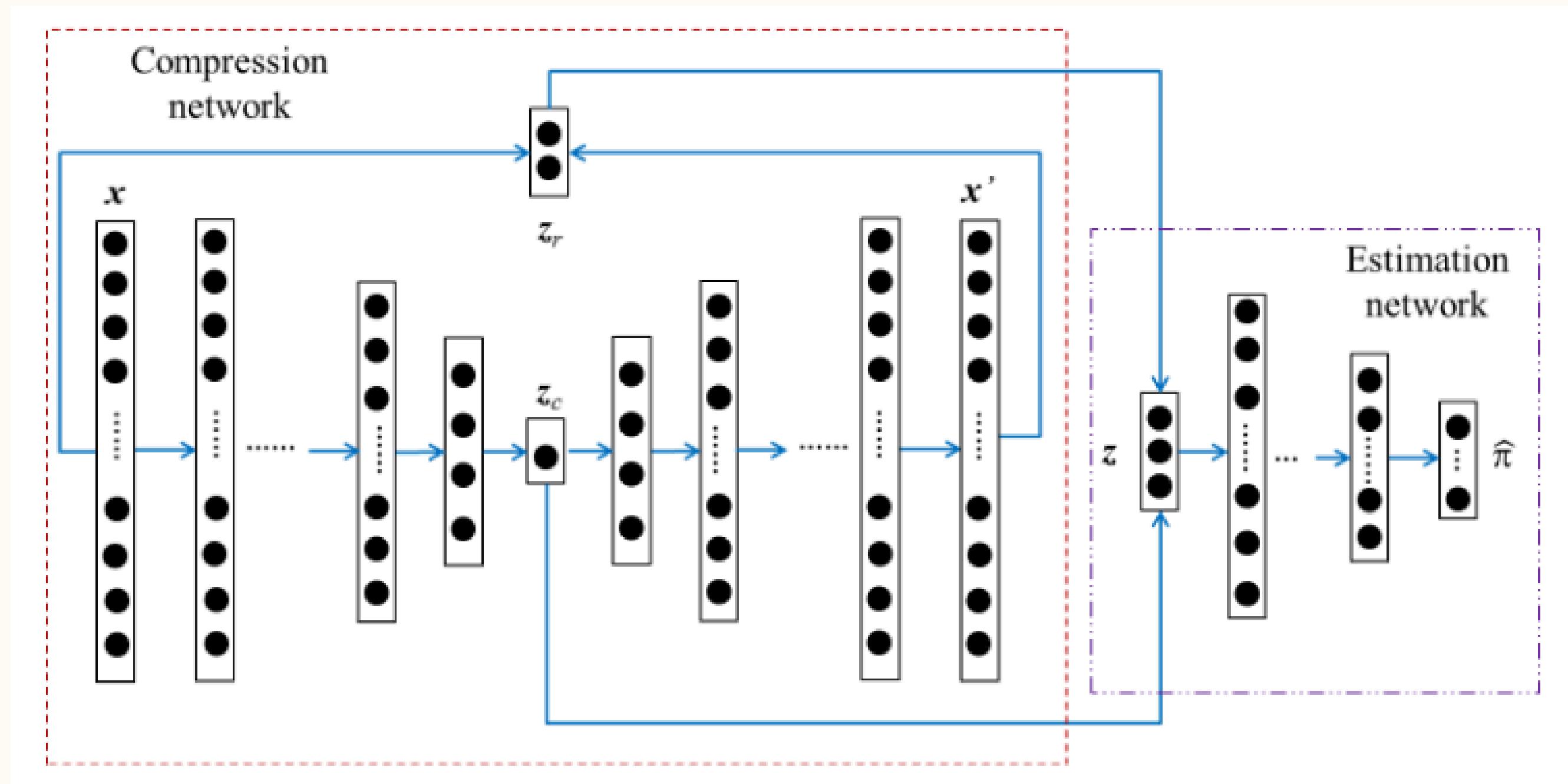


Figure 2: An overview on Deep Autoencoding Gaussian Mixture Model

Deep Autoencoding Gaussian Mixture Model for
Unsupervised Anomaly Detection
(ICLR 2018)

Deep Autoencoding Gaussian Mixture Model (DAGMM)

Architecture



$$\mathbf{z}_c = h(\mathbf{x}; \theta_e)$$

$$\mathbf{x}' = g(\mathbf{z}_c; \theta_d)$$

$$\mathbf{z}_r = f(\mathbf{x}, \mathbf{x}')$$

$$\mathbf{z} = [\mathbf{z}_c, \mathbf{z}_r]$$

Deep Autoencoding Gaussian Mixture Model (DAGMM)

Loss function and update equations

Estimation Network Output

$$\mathbf{p} = MLN(\mathbf{z}; \theta_m)$$

$$\hat{\gamma} = \text{softmax}(\mathbf{p}),$$

EM update

$$\hat{\phi}_k = \sum_{i=1}^N \frac{\hat{\gamma}_{ik}}{N} \quad \hat{\mu}_k = \frac{\sum_{i=1}^N \hat{\gamma}_{ik} \mathbf{z}_i}{\sum_{i=1}^N \hat{\gamma}_{ik}}$$

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^N \hat{\gamma}_{ik} (\mathbf{z}_i - \hat{\mu}_k)(\mathbf{z}_i - \hat{\mu}_k)^T}{\sum_{i=1}^N \hat{\gamma}_{ik}}$$

Sample Energy

$$E(\mathbf{z}) = -\log \left(\sum_{k=1}^K \hat{\phi}_k \frac{\exp \left(-\frac{1}{2} (\mathbf{z} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (\mathbf{z} - \hat{\mu}_k) \right)}{\sqrt{|2\pi \hat{\Sigma}_k|}} \right)$$

Loss Function

$$J(\theta_e, \theta_d, \theta_m) = \frac{1}{N} \sum_{i=1}^N L(\mathbf{x}_i, \mathbf{x}'_i) + \frac{\lambda_1}{N} \sum_{i=1}^N E(\mathbf{z}_i) + \lambda_2 P(\hat{\Sigma})$$

Replication in DAGMM ICLR 2018

Method	KDDCUP		
	Precision	Recall	F_1
OC-SVM	0.7457	0.8523	0.7954
DSEBM-r	0.1972	0.2001	0.1987
DSEBM-e	0.7369	0.7477	0.7423
DCN	0.7696	0.7829	0.7762
GMM-EN	0.1932	0.1967	0.1949
PAE	0.7276	0.7397	0.7336
E2E-AE	0.0024	0.0025	0.0024
PAE-GMM-EM	0.7183	0.7311	0.7246
PAE-GMM	0.7251	0.7384	0.7317
DAGMM-p	0.7579	0.7710	0.7644
DAGMM-NVI	0.9290	0.9447	0.9368
DAGMM	0.9297	0.9442	0.9369

Method	Arrhythmia		
	Precision	Recall	F_1
OC-SVM	0.5397	0.4082	0.4581
DSEBM-r	0.1515	0.1513	0.1510
DSEBM-e	0.4667	0.4565	0.4601
DCN	0.3758	0.3907	0.3815
GMM-EN	0.3000	0.2792	0.2886
PAE	0.4393	0.4437	0.4403
E2E-AE	0.4667	0.4538	0.4591
PAE-GMM-EM	0.3970	0.4168	0.4056
PAE-GMM	0.4575	0.4823	0.4684
DAGMM-p	0.4909	0.4679	0.4787
DAGMM-NVI	0.5091	0.4892	0.4981
DAGMM	0.4909	0.5078	0.4983

Model	Precision	Recall	F1-score
KDD-Cup	0.9600	0.9000	0.9300
Arrhythmia Version 1 (HP)	0.6826	0.5000	0.5772
Arrhythmia Version 2 (HP)	0.6610	0.4862	0.56028

Table 1: Performance metrics for Arrhythmia and KDD-Cup datasets

Image from the paper

Gaps in Current Research

DAGMM outperformed the SOTA back then by a large margin

All architectures are using separate steps for mapping to low dimensions and making estimations

SOTA for Image Anomaly Detection

PatchCore, 2021: Pre-trained CNN gives latent memory bank + Nearest Neighbour Search in Memory Bank

CFAE, 2023: Pre-trained CNN gives latent features + Feature-wise Dissimilarity using euclidean distance

SOTA for Audio Anomaly Detection

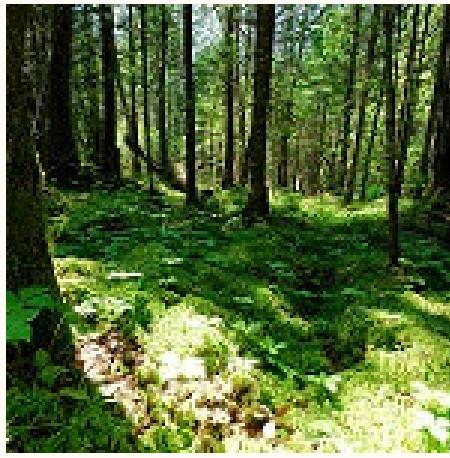
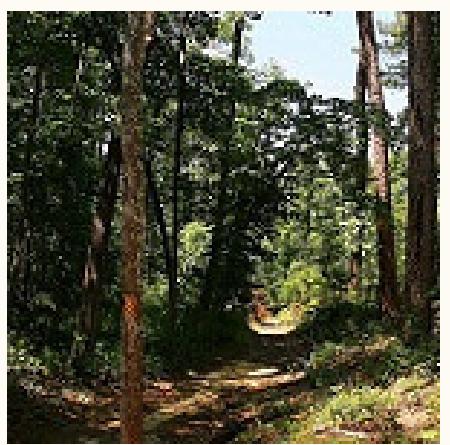
SSAD, 2021 : Pre-trained audio encoder gives latent embeddings + Nearest Neighbour Search in training embedding bank

DCASE 2021 Baseline: Pre-trained auto-encoder gives latent features + Dissimilarity using Euclidean distance

Convolutional DAGMM Datasets

Intel Image Classification

Image Scene Classification of Multiclass



Forests

vs

Others(Mountain, Building etc.)



MVTec AD

The MVTEC Anomaly Detection Dataset



Good bottle

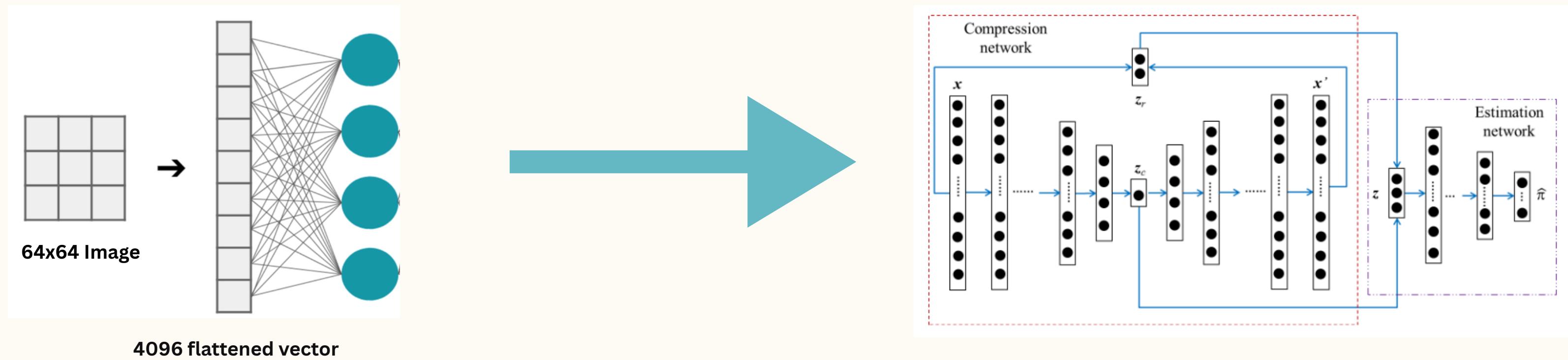
vs

Broken bottle



Convolutional DAGMM

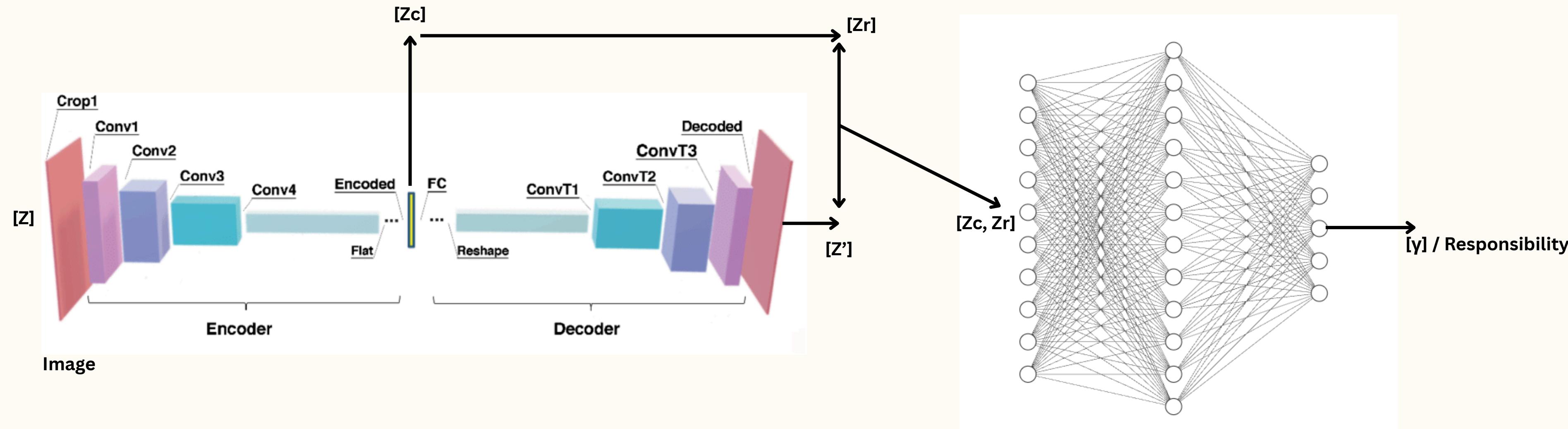
Preliminary Approach



Results: Extremely poor AUROC, precision and recall.

Convolutional DAGMM

Improved Architecture



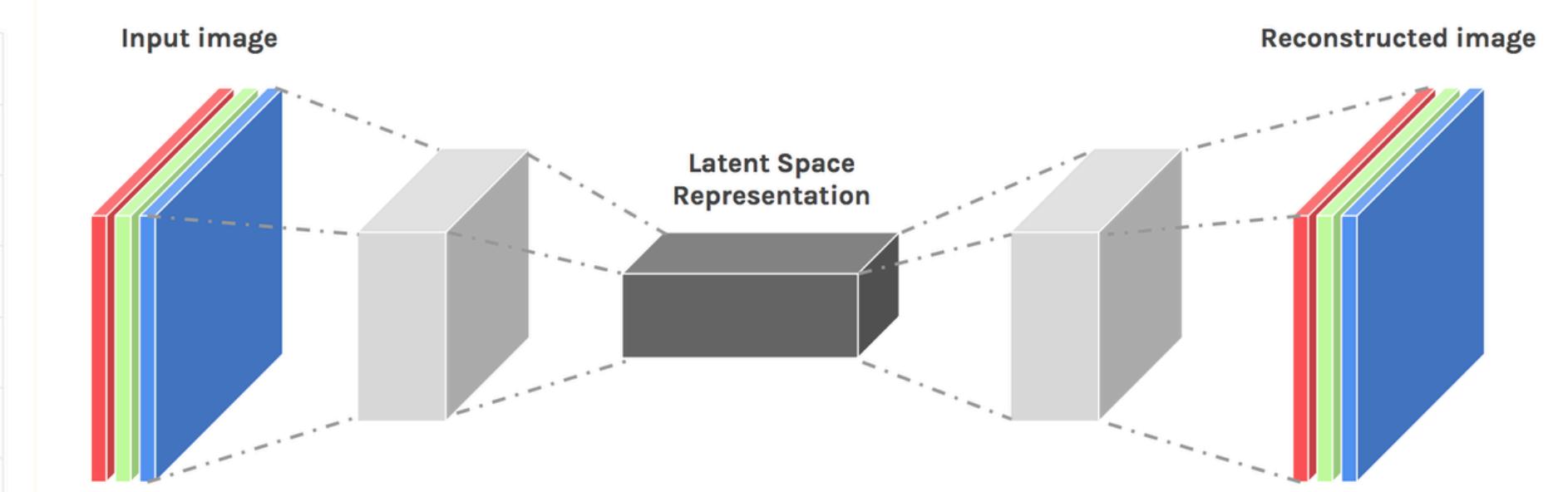
Compression Network

Estimation Network

Convolutional DAGMM

Framework 1 - Normal Conv Setup

Layer	Depth	Height	Width	Filter Height	Filter Width
Input	3	64	64	–	–
Conv1 (enc1)	32	32	32	4	4
Conv2 (enc2)	64	16	16	4	4
Conv3 (enc3)	128	8	8	4	4
Conv4 (enc4)	256	4	4	4	4
Latent FC	–	–	–	–	–
Deconv4 (dec4)	128	8	8	4	4
Deconv3 (dec3)	64	16	16	4	4
Deconv2 (dec2)	32	32	32	4	4
Deconv1 (dec1)	3	64	64	4	4



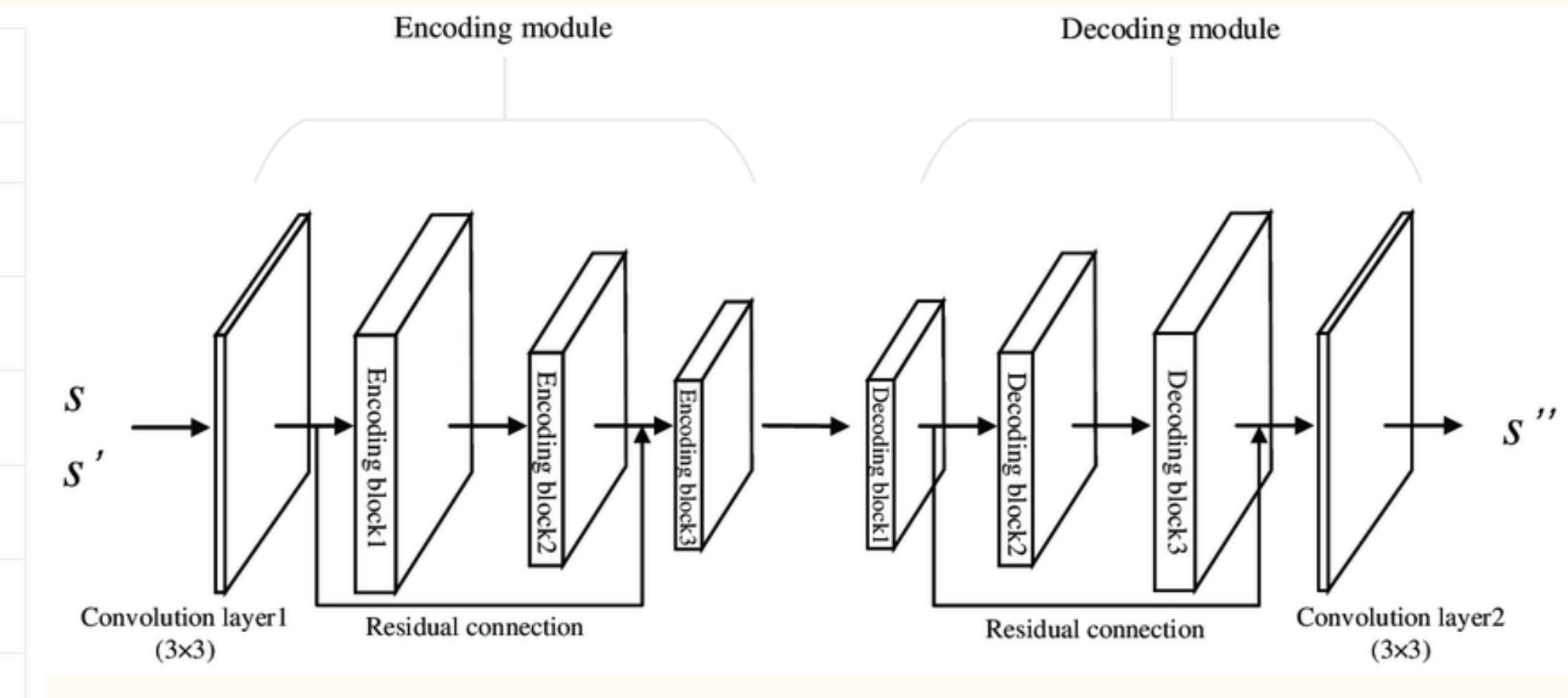
Model	Precision	Recall	AUROC
Intel Image Classification (Ours)	0.75	0.70	0.87
Intel Image Classification (SOTA)	-	-	-
MVTec AD (Ours)	0.40	0.50	0.705
MVTec AD (SOTA - GLASS)	-	-	100

Table 2: Performance metrics for Conv Encoder As Compression Network

Convolutional DAGMM

Framework 2 - Residual Conv Setup

Stage	Operation	In → Out Size	Filter HxW (layer 1 & 2)	Shortcut
Input	—	$3 \times 64 \times 64$	—	—
enc1	ResConvBlock	$3 \times 64 \times 64 \rightarrow 32 \times 32 \times 32$	4x4, str 2, p 1 & 3x3, str 1, p 1	1x1 str 2 convBN
enc2	ResConvBlock	$32 \times 32 \times 32 \rightarrow 64 \times 16 \times 16$	same	same
enc3	ResConvBlock	$64 \times 16 \times 16 \rightarrow 128 \times 8 \times 8$	same	same
enc4	ResConvBlock	$128 \times 8 \times 8 \rightarrow 256 \times 4 \times 4$	same	same
Flatten	Flatten	$256 \times 4 \times 4 \rightarrow 4096$	—	—
fc_latent	Linear	$4096 \rightarrow \text{latent_dim}$	—	—
fc_expand	Linear + reshape	$\text{latent_dim} \rightarrow 256 \times 4 \times 4$	—	—
dec4	ResDeconvBlock	$256 \times 4 \times 4 \rightarrow 128 \times 8 \times 8$	4x4, str 2, p 1 & 3x3, str 1, p 1	Upsample2 + 1x1 convBN
dec3	ResDeconvBlock	$128 \times 8 \times 8 \rightarrow 64 \times 16 \times 16$	same	same
dec2	ResDeconvBlock	$64 \times 16 \times 16 \rightarrow 32 \times 32 \times 32$	same	same
dec1 (out)	ResDeconvBlock + Sigmoid	$32 \times 32 \times 32 \rightarrow 3 \times 64 \times 64$	same	same

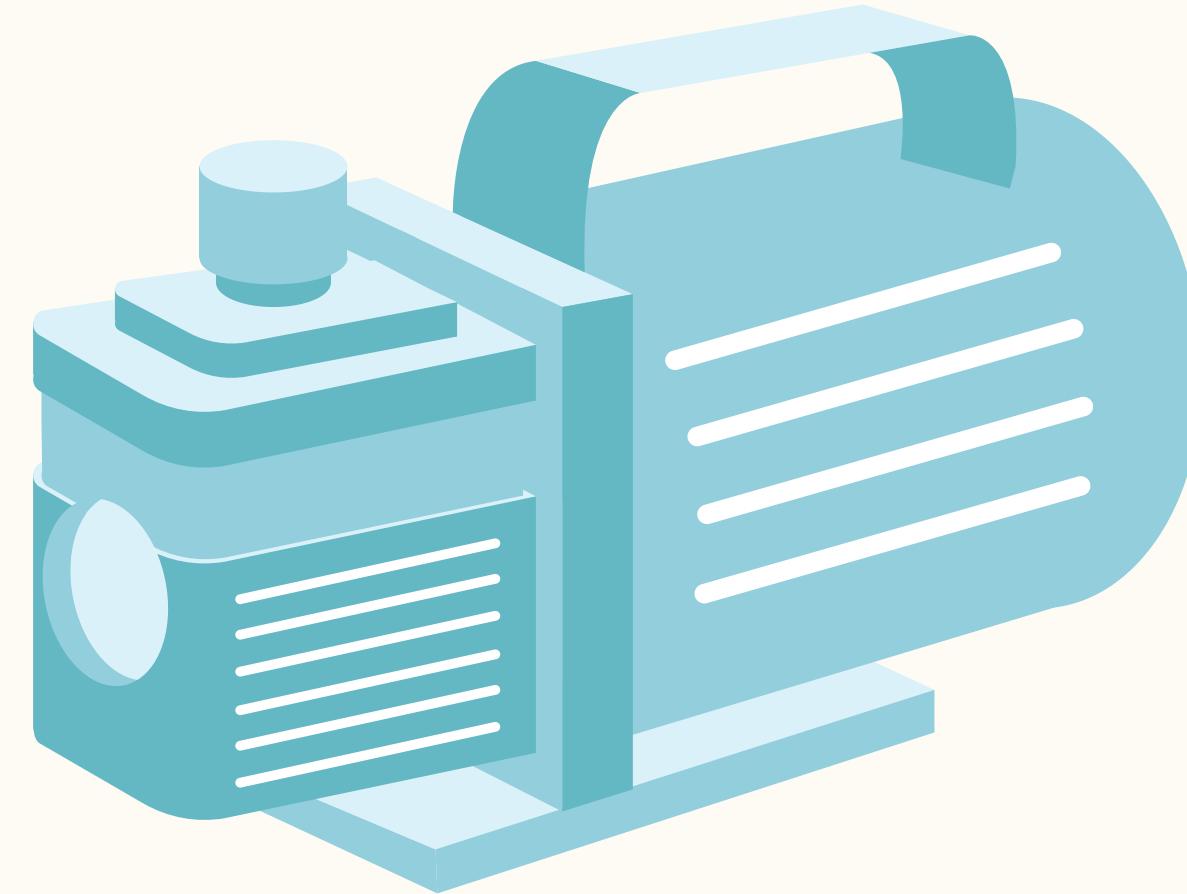


Model	Precision	Recall	AUROC
Intel Image Classification (Ours)	0.58	0.55	0.829
Intel Image Classification (SOTA)	-	-	-
MVTec AD (Ours)	0.28	0.30	0.70
MVTec AD (SOTA - GLASS)	-	-	100

Table 3: Performance metrics for Residual Conv Encoder As Compression Network

The audio anomaly challenge

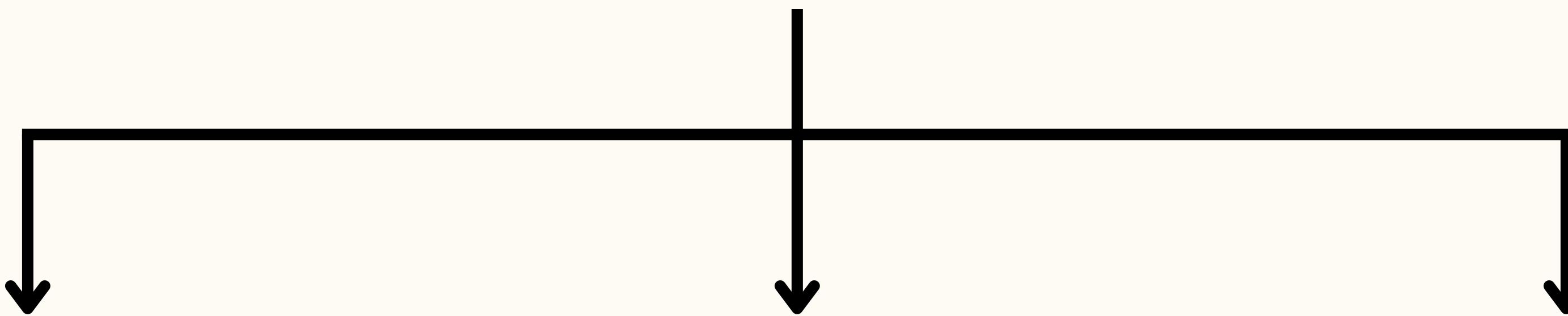
Dataset



MIMII Pump Sound Dataset

MIMII Pump is a labeled audio dataset of industrial pumps operating normally and under various faults, designed for anomaly-detection research.

Approaches for Audio



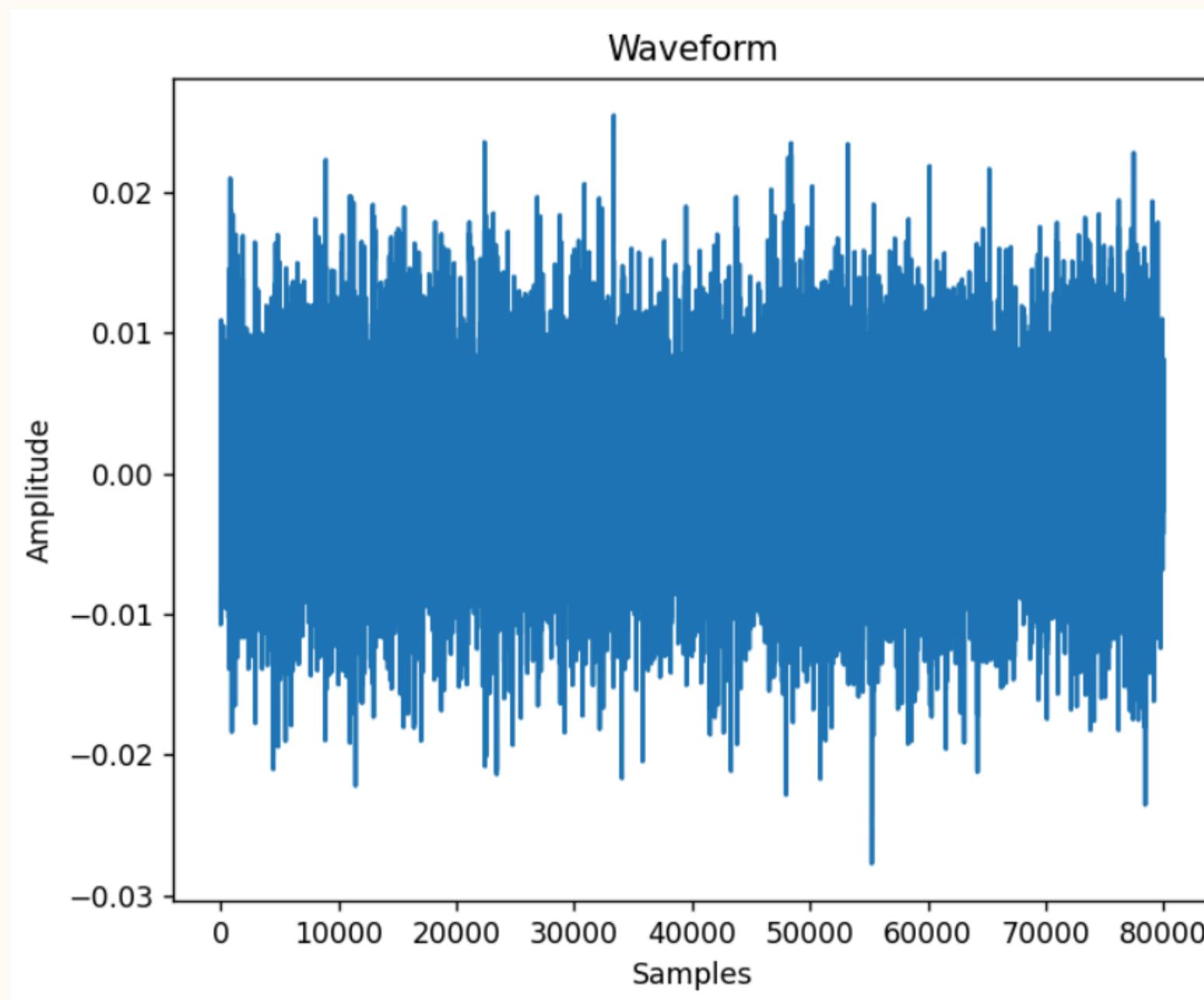
Flattening the dataset and using DAGMM

Flattening MEL scaled dataset and using DAGMM

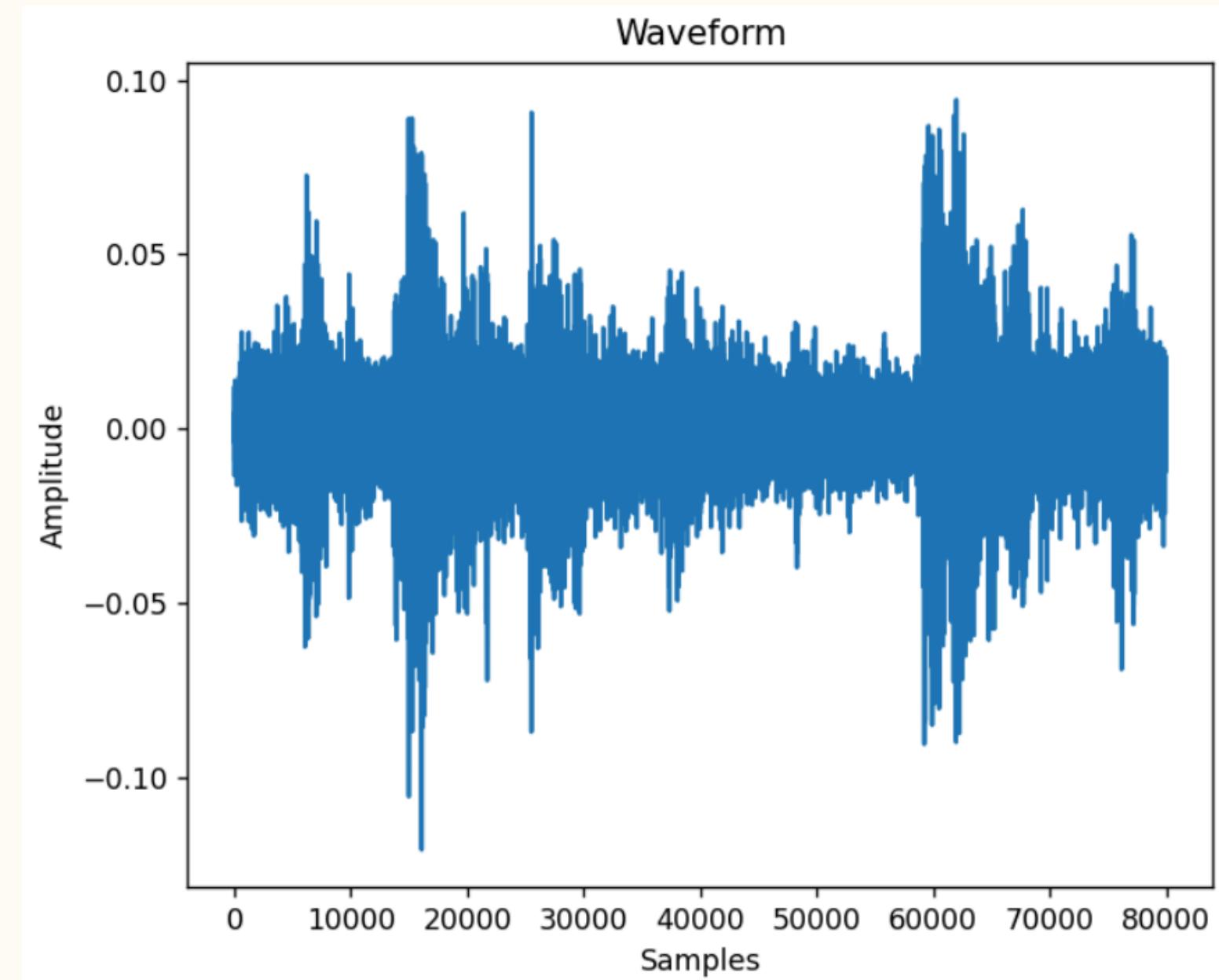
Using CNN + DAGMM on MEL scaled dataset

DAGMM for Audio Datasets

Normal

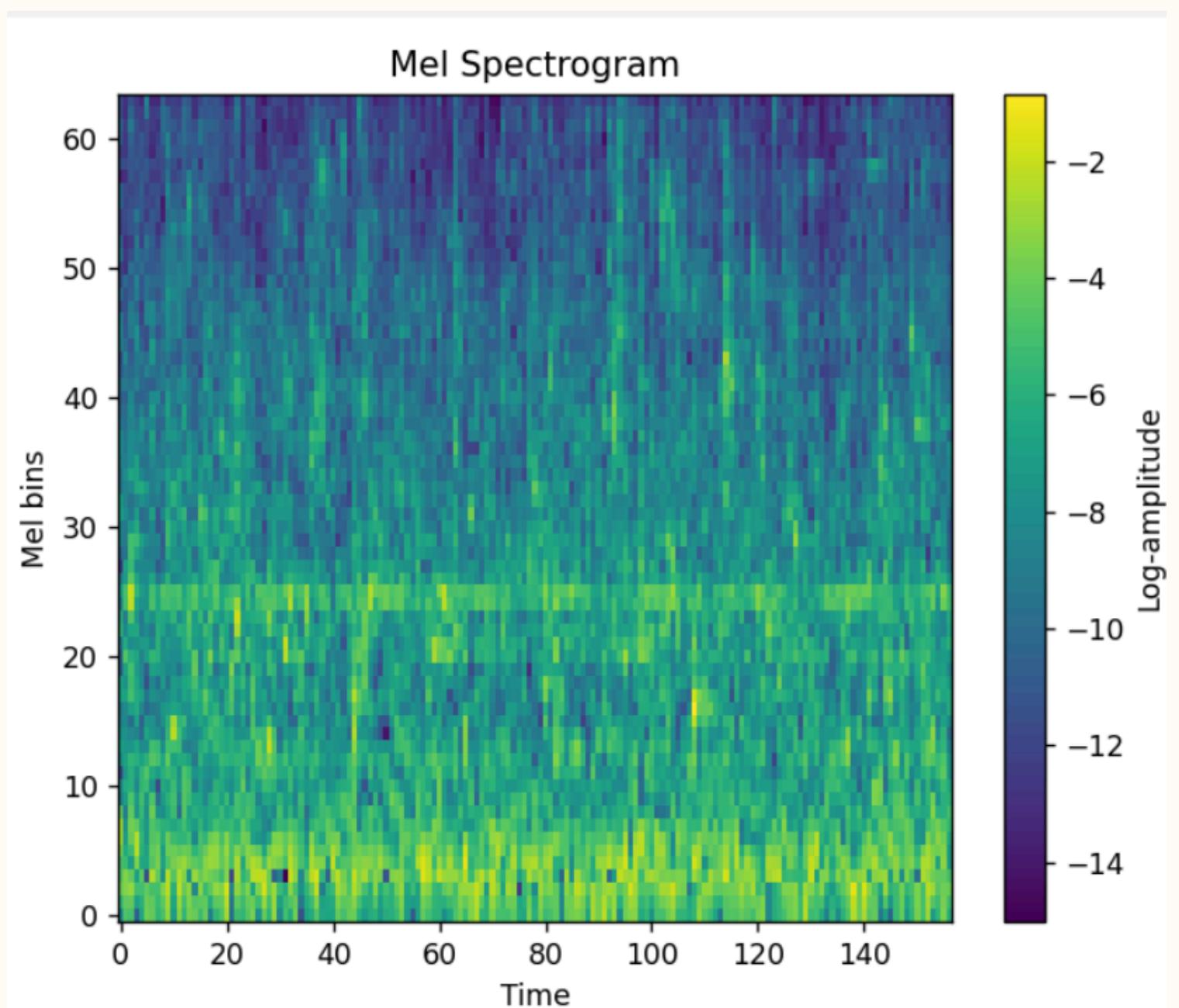


Abnormal

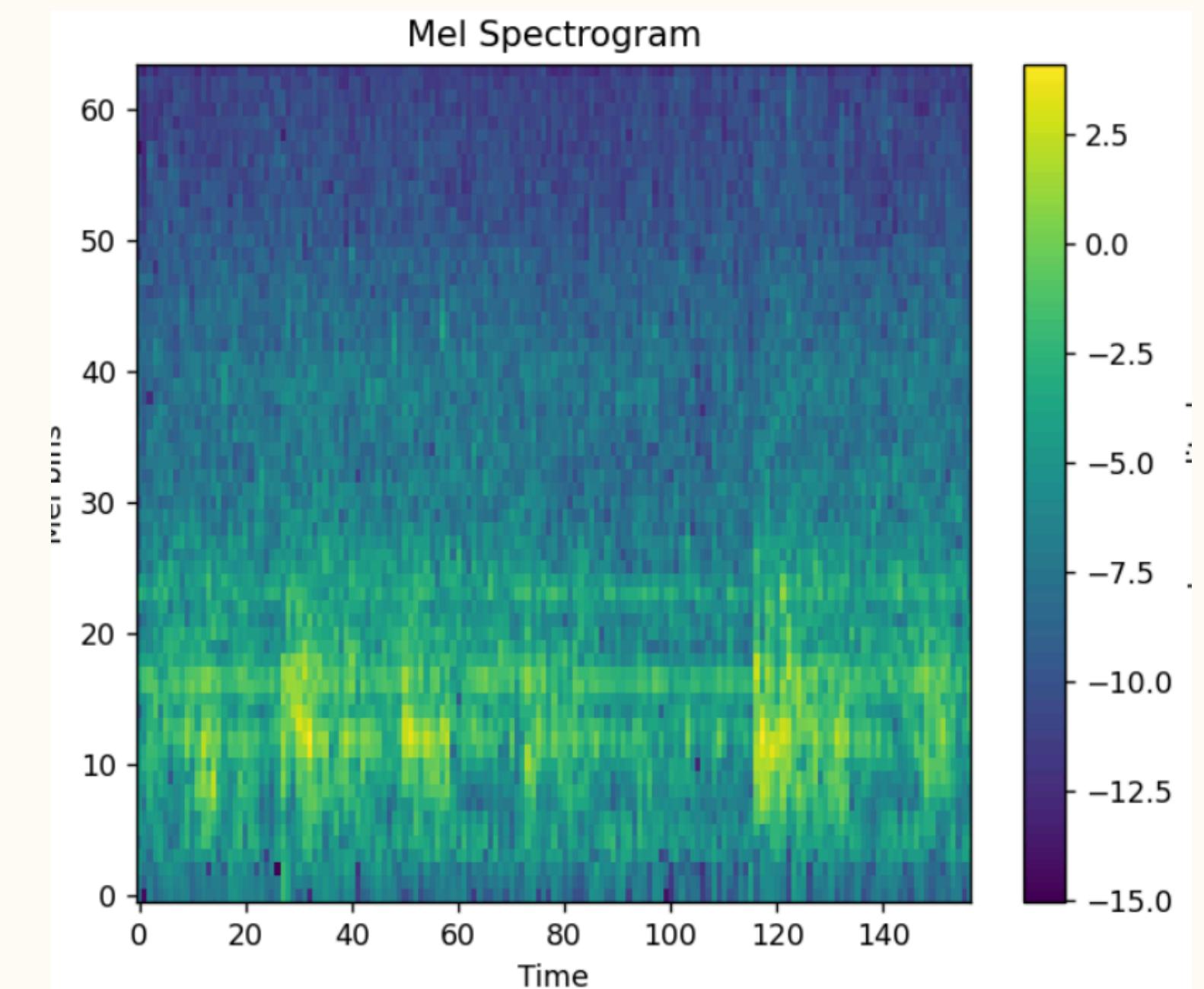


MEL Spectrogram for Audio Datasets

Normal



Abnormal



DAGMM for Audio

Model	Precision	Recall	F1-Score	AUC	pAUC (FPR ≤ 0.1)
Audio (Ours)	0.6410	0.6944	0.6667	0.8050	0.6613
Audio Mel (Ours)	0.6667	0.6582	0.6624	0.7464	0.6371
CNN Mel (Ours)	0.6010	0.4216	0.5057	0.4532	0.4098
Competition Model (SOTA)	-	-	-	0.9000	0.7100

Table 4: Performance metrics for Audio Anomaly Detection Models

Parameter	Audio (Ours)	Audio Mel (Ours)
Sample Rate	16000	16000
Time (sec)	5	5
Epochs	40	30
Activation	tanh	tanh
Threshold	0.7	0.7
Latent Space	2	2
GMM/Nbins/Hop Length	GMM: 2	Nbins: 64, Hop Length: 512, GMM: 2

Table 5: Hyperparameters for Audio Anomaly Detection Models

Conclusion

Implemented the full code of the DAGMM architecture

Explored multiple ways to extend the DAGMM architecture to images and audio

Got considerable accuracy in while testing on multiple datasets

Future Prospects

Extend to Multi Modal Data

Try different models for compression and estimation
networks like GANs + feature disimilarity

Try even more architectures and hyper parameters on
the models currently developed