NORD-F: Addressing Near-OoDs for Robust Classification through Disentanglement Representation Learning for Fine Grained Datasets

A Project Report

submitted by

ISHIKA SHARMA

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

ELECTRONICS AND COMMUNICATION ENGINEERING
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

July 2024

THESIS CERTIFICATE

This is to certify that the thesis titled **NORD-F: Addressing Near OoDs for Robust Classification through Disentanglement Representation Learning with Fine Grained Data**, submitted by **Ishika Sharma**, to the Indraprastha Institute of Information Technology, Delhi, for the award of the degree of **MASTERS OF TECHNOLOGY**, is a bonafide record of the research work done by her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr.Ranjitha Prasad
Thesis Supervisor
Assistant Professor
Dept.of Electronics and Communication
IIIT Delhi, 110020

Place: New Delhi

Date: 29th July 2024

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to Dr. Ranjitha Prasad for her unwavering guidance, exceptional support, and remarkable supervision throughout the course of this research work. Her expertise and insightful feedback have been invaluable in shaping the direction and quality of this thesis. I am profoundly grateful for her dedication and commitment to my academic development.

Also, I would like to extend my heartfelt gratitude to Ms.Vidya T and Mr.Ashish Sethi from LightMetrics Technologies Pvt. Ltd for their invaluable guidance throughout this project. Their insights and support have been instrumental in shaping this research. Additionally, I am deeply appreciative of Mr.Mithun Uliyar and Mr.Krishna A.G for giving me the opportunity to collaborate with such a distinguished team. Additionally, I would like to express my deepest appreciation to my family and friends for their unwavering care, unconditional love, and constant blessings throughout this journey. Their encouragement and belief in me have been instrumental in my success.

ABSTRACT

KEYWORDS: Disentanglement; Fine-grained; Near-OoD; Gradient Reversal

Out-of-Distribution (OoD) detection has emerged as a crucial aspect in machine learning, essential for ensuring the resilience and reliability of models deployed in real-world scenarios. Traditional methods excel at identifying far-OoDs, but struggle with near-OoDs since the differences between in-distribution and near-OoD samples are subtle. Conventional techniques of OoD detection such as confidence scores or likelihood measures often fail in the context of detecting near-OoDs. This discrepancy highlights the necessity for novel approaches for detecting near-OoDs, particularly for classification tasks in fine-grained datasets, where limited discriminative features alongside intra-class variability is a critical issue.

We explore disentangled representation learning (DRL), where we seek to extract relevant features essential for accurate classification, while disentangling irrelevant features. In this work, we assume that the OoD samples occur during inference, and hence, model is unaware of OoDs during training. Hence, there is an evident shift between the training and test distributions. An important question to pose in this context is the following: *Can near-OoD detection in such a context be expressed as a problem of domain adaptation?*.

Domain adaptation methods build the mappings between the source (training-time) and the target (test-time) domains, so that the classifier learned for the source domain can be used on the target domain during inference. In this work, we employ domain adaptation based gradient reversal layer for vector-wise disentanglement of feature vectors into class-specific and class-invariant features. We propose the novel NORD-F framework, which consists of a classifier branch, a encoder-decoder based DRL branch and a variation branch.

Using experiments on fine-grained datasets such as Stanford Dogs, FGVC-Aircraft, etc, we demonstrate that the proposed method outperforms OoD-aware baselines in terms of several OoD metrics. Further, using TSNE visualization, we illustrate that our approach disentangles the feature representation as class-invariant and class-specific features. Hence, by leveraging disentangled representation learning and insights from domain adaptation, our approach identifies near-OoDs ensuring the model's awareness towards OoD samples. This research contributes to the advancement of OoD detection methodologies, offering an efficient framework suited to address the challenges of fine-grained datasets.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ID	In-distribution
OoD	Out-of-distribution
DRL	Disentanglement Representation Learning
GRL	Gradient Reversal Layer

CHAPTER 1

INTRODUCTION

Out-of-Distribution (OoD) detection is crucial in ML models used in safety critical domains such as healthcare, automobile industry, finance, etc. OoD detection helps identify when the data deviates from the in distribution samples, and identifying adversarial attacks where inputs are designed to mislead the model. Furthermore, regulatory standards or ethical considerations may require that models operate within well-defined conditions and reject data outside of those conditions, and OoD detection plays a crucial role here. Most of the literature on OoD detection handle far-OoDs, which focuses on identifying data that is significantly different from the training distribution. For instance, the ML models are trained on the MNIST dataset, while a sample from the Fashion-MNIST dataset is used as an OoD Ren *et al.* (2019) or synthetic OoD samples are generated using GANs Mukhoti *et al.* (2023). Such techniques handle substantial deviations of far-OoDs in feature space as compared to in-distribution (ID) samples, but falter when confronted with near-OoD instances which semantically look very similar to the ID data but have a different label Mukhoti *et al.* (2023). The importance of near-OoD detection is underscored by its implications across various domains. For instance, in medical imaging, distinguishing between different types of skin lesions requires a model not only to identify common classes but also to recognize rare anomalies that may signal critical conditions Mehta *et al.* (2022). Similarly, in autonomous vehicles, accurately identifying pedestrians under various lighting and environmental conditions is crucial for ensuring safety.

In this work, we propose a novel NORD-F framework, for discerning near-OOD samples from ID samples. In this chapter, we introduce the different modules that have been incorporated in this work.

1.1 Out-of-Distribution Detection

There are several approaches to handling OoD samples. Furthermore, one may choose to obtain OoD generalization Liu *et al.* (2021*a*) or perform OoD detection Yang *et al.* (2024). Both are two related but distinct challenges in machine learning, although both deal with handling data that deviates from the distribution on which a model was trained.

OoD generalization Liu *et al.* (2021*a*) methods aim to enhance the model's ability to generalize to unseen environments or dataset shifts. These methods focus on improving model performance on OoD samples by incorporating training strategies that foster model adaptability and robustness. Techniques such as data augmentation Hendrycks *et al.* (2019), ensemble methods Segu *et al.* (2023), and domain generalization technique Wang *et al.* (2022*a*) are commonly utilized to achieve this objective.

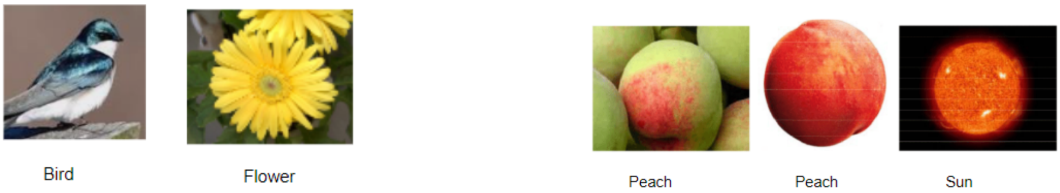
In contrast, OoD detection methods Yang *et al.* (2024, 2022) serve the objective of detecting instances that deviate significantly from the training distribution. These methods are primarily concerned with scenarios where OoDs occur during inference, and hence, models' outputs or intermediate representations are analyzed to determine if a given unseen sample is OoD. Various techniques such as confidence scores Hendrycks and Gimpel (2016), distance-based metrics like Mahalanobis distance Lee *et al.* (2018), adversarial training based Bitterwolf *et al.* (2020) and reconstruction loss Zhou (2022); Jiang *et al.* (2023) are employed for OoD detection. These techniques helps the model to avoid incorrect predictions and enhance model reliability Yang *et al.* (2022). In this work, we assume that an unseen sample is OoD and hence, we focus on OoD detection rather than generalization.

1.1.1 Understanding Far and Near OoDs

When exploring the concept of OoD instances, distinguishing between near and far variations provides valuable insights into data representation complexities.
Far-OoD: Far-OoD instances represent samples that are perceptually and semantically dissimilar to the training data. These instances exhibit entirely different features and labels compared to the training distribution.

Near-OoD: Near out-of-distribution (OoD) instances present a scenario where the samples exhibit perceptual similarities to the training data but possess semantic differences. These instances are characterized by subtle deviations that closely resemble the training distribution, making them challenging to identify using traditional OoD detection methods. They may share overlapping features with the training data, yet exhibit dissimilar labels Mukhoti *et al.* (2023), necessitating a fine-grained analysis to discern their out-of-distribution nature.

Unlike near-OoD instances, far-OoD instances can be readily identified due to their stark dissimilarity to the training data, resulting in a relatively simple decision boundary between ID and OoD samples. Consequently, detecting far-OoD instances may require less complex algorithms compared to near-OoD detection, as their distinctive features facilitate straightforward discrimination. Due to the complex decision boundary between near-OoD and ID samples, detecting and accurately classifying these instances requires sophisticated OoD detection algorithms Zhang *et al.* (2023*b*) capable of nuanced discrimination.



(a) Example of Far-OoD

(b) Example of Near-OoD

Figure 1.1: In Fig a) both the images are both semantically and perceptually different but in Fig b) peach and sun look perceptually similar which can affect model's detection capability

1.2 Disentanglement Representation Learning

Disentanglement representation learning fundamentally revolves around the idea of extracting underlying factors of variation within data leading to improved interpretability and separability. By disentangling factors of variation, such as object identity, pose, or lighting conditions in images, models can learn more robust and transferable representations, leading to more efficient learning algorithms and better performance in downstream tasks.

Based on the structure of disentangled representations, DRL methods are categorized into two groups, dimension-wise and vector-wise methods Wang *et al.* (2022*b*). Dimension-wise disentanglement incorporates generative models to ensure that individual dimensions within the latent space represents one specific attribute or factor. This leads to obtaining the degree to which different dimensions in the latent space can be mapped to separate and meaningful factors of variation in the data Xiao *et al.* (2017). Despite its potential benefits, dimension-wise disentanglement is challenging as it requires labels for interpretable dimensions alongside relevant evaluation metrics Locatello *et al.* (2019). Among different vector-wise disentanglement techniques Lee *et al.* (2021), we are interested in techniques that explicitly model the intra-class variance and disentangle the intra-class invariance Lin *et al.* (2018).

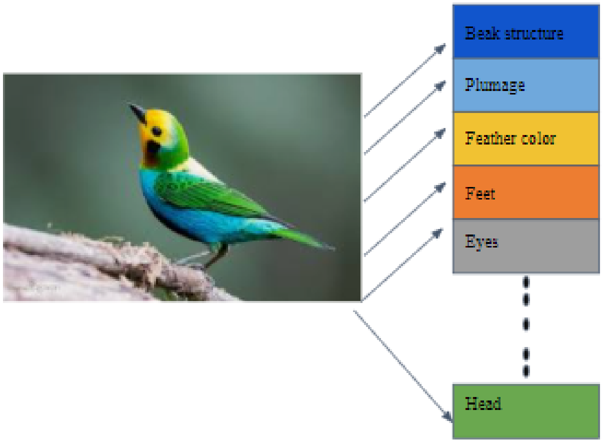


Figure 1.2: Disentanglement feature vector of a bird's image

1.3 Domain Adaptation and OoD detection

Domain adaptation is a crucial technique in machine learning that aims to address the challenge of applying models trained on one domain (source domain) to a different but related domain (target domain). One of the notable approaches in domain adaptation is the use of GRL, which plays a significant role in adversarial domain adaptation. GRL functions by reversing the gradients during back-propagation, encouraging the model to learn domain-invariant features by simultaneously minimizing the task-specific loss and maximizing the domain classification loss Ganin and Lempitsky (2015). This technique effectively bridges the gap between the source and target domains, enhancing the model's generalization capabilities. The principles of GRL are particularly relevant in the context of OoD detection. By promoting domain-invariant feature learning, models become more robust to variations in the input data, improving their ability to identify and handle OoD samples. This connection underscores the importance of domain adaptation approaches like application of GRL in developing reliable and adaptable machine learning systems that perform well even when faced with data from different distributions.

1.3.1 Significance of GRL In Our Work

GRL has emerged as a pivotal component in domain adaptation and domain generalization tasks Zhang *et al.* (2022), serving to mitigate domain shift and promote domain-invariant representations.

In the context of OoD detection, GRL has been integrated into frameworks to enhance model robustness against distribution shifts and improve OoD detection performance. Understanding the role of GRL in domain adaptation and its implications for OoD detection methodologies is essential for designing effective and adaptable models capable of handling real-world data distributions.

By synthesizing diverse strands of knowledge, we developed a novel framework NORD-F for near-OoD detection. Our approach leverages DRL combined with domain adaptation techniques to effectively identify near-OoDs. GRL, which reverses the gradient direction during the backward pass. This counteracts the typical deep neural network's tendency to learn clustered and relevant features for the downstream task. Instead, due to the GRL, the network begins to learn irrelevant features of the image and ultimately we segregate the most relevant and irrelevant features for our task.

In the context of out-of-distribution (OoD) detection, the Gradient Reversal Layer (GRL) has been successfully integrated into various frameworks to enhance model robustness against distribution shifts and enhance OoD detection capabilities. Grasping the significance of GRL in domain adaptation and its impact on OoD detection methodologies is crucial for designing models that can effectively manage real-world data distributions.

By synthesizing different areas of research, we have developed an innovative pipeline for near-OoD detection. Our approach combines disentangled representation learning with domain adaptation techniques to proficiently identify near-OoDs. GRL, by reversing the gradient direction during backpropagation, counteracts the natural tendency of deep neural networks to focus on clustered and task-relevant features. Instead, this process enables the network to learn a broader range of features, including those that are typically deemed irrelevant. This allows us to distinguish between the most pertinent and extraneous features for our specific task, enhancing the model's ability to detect near-OoD instances.

1.4 Contributions

We propose the NORD-F framework which is a novel architecture for near-OoD detection in fine-grained datasets. The architecture consists of the following:

- Variation branch: This branch encodes the class-irrelevant information that complements the image representation. This branch employs domain adaptation based principles using GRL.
- Reconstruction branch: This branch ensures that disentangled features together lead to input image reconstruction, and hence, image information is preserved.
- Classification branch: This branch extracts the discriminative features of the image sample.

Broadly, the contributions of this thesis are as follows:

- Addressed the challenge of detecting near-OoD samples through disentanglement representation learning where we extract the most relevant features and abstain the features which does not contribute to the classification task.
- Introduced a novel approach for OoD detection in fine-grained environment using GRL application inspired from domain adaptation methods.
- We evaluate the proposed framework on fine-grained datasets such as Stanford Dogs, FGVC-Aircraft.

1.4.1 Notations

Small letters denote scalars, boldface small letters denote vectors. $\mathbf{I}$ denotes an identity matrix whose size is as per context. The ℓ_2-norm of a vector $\mathbf{x}$ is denoted as $||\mathbf{x}||$. $\mathcal{P}$ represents sets and $|\mathcal{P}|$ represents size of the set.

CHAPTER 2

RELATED WORKS AND PRELIMINARIES

In this chapter, we delve into the existing literature surrounding the key topics pertinent to our research. We explored various aspects such as DRL and the sub-types, fine-grained analysis for OoD detection, GRL technique for disentanglement purpose, etc. By reviewing prior research, we aim to establish a comprehensive understanding of the theoretical foundations, methodological approaches that can help to address the gaps in the recent developments.

2.1 Existing Out-of-Ditribution Detection Methods

We broadly classify the relevant OoD detection methods as post-hoc, training-based, ViT-based, training with outlier exposure strategy, OoD data generation technique and Bayesian methods.

Post-Hoc Methods: Post hoc methods for OoD detection are techniques applied after model training to identify out-of-distribution samples. These methods modify model outputs or decision thresholds to improve the detection of OoD samples without re-training the model.

MSP: The first originated baseline for OoD detection is Maximum Softmax Probability (MSP) which uses the highest softmax score from a classifier as a confidence measure. If this maximum probability is below a certain threshold, the input is considered as OoD.Hendrycks and Gimpel (2016).

Energy-based: involve using the energy score of a sample, derived from the model's output logits, to determine the likelihood of being in-distribution. Lower energy scores typically indicate in-distribution samples, while higher scores suggest OoD samples. Liu *et al.* (2020)

Temperature Scaling: adjusts the confidence scores by scaling the logits and improves the calibration of the model's predictions Guo *et al.* (2017).

Training-Based Method: These methods involve incorporating OOD detection mechanisms directly into the training process of the model.
LogitNorm: This method modifies the standard cross-entropy loss by enforcing a constant vector norm on the logits during training. As a result, networks trained with LogitNorm produce confidence scores that are distinguishable between in-distribution (ID) and out-of-distribution (OoD) data, improving the accuracy of OoD detection.Wei *et al.* (2022)

ODIN: It is a post-hoc plus training based approach that utilizes temperature scaling and input perturbation to enhance the separability between in-distribution (ID) and out-of-distribution (OoD) samples. A sufficiently large temperature smooths the softmax scores, effectively transforming them back to the logit space, which improves the distinction between ID and OoD samples.Liang *et al.* (2018) This differs from Temperature Scaling Guo *et al.* (2017) method that uses a much milder temperature.

Bayesian Methods: For OoD detection, variational autoencoders (VAEs) based Feng *et al.* (2021); Ran *et al.* (2022); Daxberger and Hernández-Lobato (2019) and other Bayesian methods like Choi *et al.* (2021) provide a robust framework for addressing OoD detection. The paper Ramakrishna *et al.* (2022) highlights the effectiveness of utilizing the latent space of β-VAEs for OoD detection in cyber-physical systems. By leveraging disentangled representations, this approach ensures that the model can effectively differentiate between ID and OoD samples, enhancing system robustness. In Zhou (2022); Osada *et al.* (2023) the authors propose using reconstruction error as a metric for OoD detection. The underlying hypothesis is that ID data is reconstructed accurately by the autoencoder, whereas OoD data will exhibit higher reconstruction errors, making it distinguishable. The *Tilted VAE* Floto *et al.* (2023) approach introduces a tilted Gaussian prior by adding a penalty term to the VAE loss function. This method has been tested for adversarial samples and distributional shifts, showing that the tilted prior helps in effectively separating OoD samples from ID data. The paper Nguyen *et al.* (2022) uses dropout neural networks leverages the inherent uncertainty estimation capabilities of dropout. During training, dropout randomly deactivates neurons, which effectively creates an ensemble of models. By analyzing the variability (uncertainty) of these predictions at inference, the model can identify samples with high uncertainty as potential OoD samples, as ID tend to produce more confident and consistent predictions.

Vision-transformer based Approaches: In Sim *et al.* (2023), the authors explore the use of Vision Transformers (ViTs) for the task of OOD detection. The self-attention mechanism in ViTs enables the model to focus on relevant regions of the image, which is crucial for identifying OOD samples that may differ subtly from in-distribution samples. They finally Used the learned representations to distinguish between in-distribution and OOD samples based on metrics Mahalanobis distance. Similarly, the study by Fort et al. Fort *et al.* (2021) explores the capabilities of ViTs for OOD detection across different modalities.

Training with Outlier Exposure: In Zhang *et al.* (2023*a*), the authors introduce Mixture Outlier Exposure (MixOE), which combines ID data with training outliers to broaden the range of OoD granularities. This approach trains the model to ensure that prediction confidence decreases linearly as the input shifts from ID to OoD which is achieved by incorporating the following objective:

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{in}}}\left[L(f(\mathbf{x}),y)\right]+\beta\mathbb{E}_{(\tilde{\mathbf{x}},\tilde{y})\sim\mathcal{D}_{\text{virtual out}}}\left[L(f(\tilde{\mathbf{x}}),\tilde{y})\right].$$

Here, $L(f(\mathbf{x}),y)$ is the cross-entropy loss between the DNN's predicted distribution $f(\mathbf{x})$ and the ground truth distribution y, and β is a weighting term. After training, the detection will be performed by thresholding the prediction confidence. Chen *et al.* (2021) demonstrate that by mining informative auxiliary OOD data, it is possible to significantly enhance OOD detection performance. Additionally, this approach improves the model's ability to generalize to previously unseen adversarial attacks.

OoD Data Generation: The paper Mukhoti *et al.* (2023) proposes the criterias for a sample to be near-OoD and far-OoD and thereby generate OoD samples through GAN network. Through this paper we got better understanding of far and near OoDs in a comprehensive manner which was defined as follows:

To classify near and far OoD sets, they utilize the Learned Perceptual Image Patch Similarity (LPIPS) metric, which correlates well with human judgment. Let ($f_{\boldsymbol{\theta}}$) represent a pre-trained convolutional network. Given two images, $\mathbf{x}_1$ and $\mathbf{x}_2$, LPIPS computes the cosine distance between feature space activations across different layers of the network $f_{\boldsymbol{\theta}}$ as follows:

$$L_{\text{LPIPS}}(\mathbf{x}_1, \mathbf{x}_2) = \sum_l \frac{1}{H_l W_l} \left\| f^l_{\boldsymbol{\theta}(\mathbf{x}_1)} - f^l_{\boldsymbol{\theta}(\mathbf{x}_2)} \right\|_2^2$$

where $f^l_{\boldsymbol{\theta}(\mathbf{x}_1)}$ and $f^l_{\boldsymbol{\theta}(\mathbf{x}_2)} \in \mathbb{R}^{H_l \times W_l \times C_l}$ are the feature space representations from inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ in layer l. They used LPIPS with a VGG network. For semantic similarity, they used mutual information (MI) $I[y, \boldsymbol{\theta}|\mathcal{D}, \mathbf{x}]$ to quantify the information gain about the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ upon seeing a sample $\mathbf{x} \in \mathcal{D}$. For computational feasibility, a pre-trained deep ensemble approximates MI as:

$$L_{\text{MI}}(\mathbf{x}) = \hat{I}[y, \boldsymbol{\theta}|\mathcal{D}, \mathbf{x}] \approx H[p(y|\mathbf{x}, \boldsymbol{\theta})] - \frac{1}{T}\sum_{t=1}^{T} H[p(y|\mathbf{x}, \boldsymbol{\theta}_t)]$$

where $p(y|\mathbf{x}, \boldsymbol{\theta}_t)$ is the softmax distribution of the t-th network in an ensemble of T networks. $H[\cdot]$ denotes the entropy of a distribution. We use L_{MI} to measure semantic similarity to the training distribution, with low L_{MI} indicating similarity and high L_{MI} indicating dissimilarity. This inspired us to explore more about Mutual information in a VAE like set-up. In the context of VAEs, mutual information can be leveraged to enhance disentanglement by enforcing a semantic similarity loss. This concept is explored in the paper Rodriguez (2021). The objective is to maximize the mutual information between the latent variables and the observed data while ensuring that the latent variables capture distinct factors of variation. Also in the paper Liu *et al.* (2021*b*), it discusses enhancements to the Evidence Lower Bound (ELBO) in semi-supervised settings to improve classification accuracy. The approach involves modifying the ELBO to include a classification term, ensuring that the learned representations are both generative and discriminative. Vernekar *et al.* (2019) generates boundary samples in the low-density region. Du *et al.* (2022) proposed generating synthetic outliers from the regions of the feature space that have low likelihood.

In general, these papers gave a clear understanding about OoDs and various methodologies involved for OoD detction.

2.2 Disentanglement Representation Learning

Disentanglement representation learning, a fundamentally revolves around the idea of extracting underlying factors of variation within data in a way that makes them interpretable and separable. With reference to the paper Wang *et al.* (2022*b*) we analysed the following classifications of DRL:

Based on the structure of representations, DRL can be classified into two categories:

- **Dimension-wise DRL**: It focuses on disentangling each dimension independently, potentially leading to more granular control over the learned representations. Such representations provide insights into the specific aspects of the data that contribute to variations in the output Chen *et al.* (2016). Training for dimension-wise DRL often requires labels of these dimensions, and these may not be known.
- **Vector-wise DRL**: It focuses on capturing more complex relationships between features. This method of DRL is capable of considering interactions among multiple dimensions simultaneously, allowing for the extraction of higher-level concepts. It particularly emphasizes on the holistic structure of the data, enabling the learning of more abstract and generalized representations. For instance, the disentanglement is possible as coarse-grained and fine-grained generative factors, time invariant (content) and a time-varying (pose) Denton *et al.* (2017); Lee *et al.* (2021).

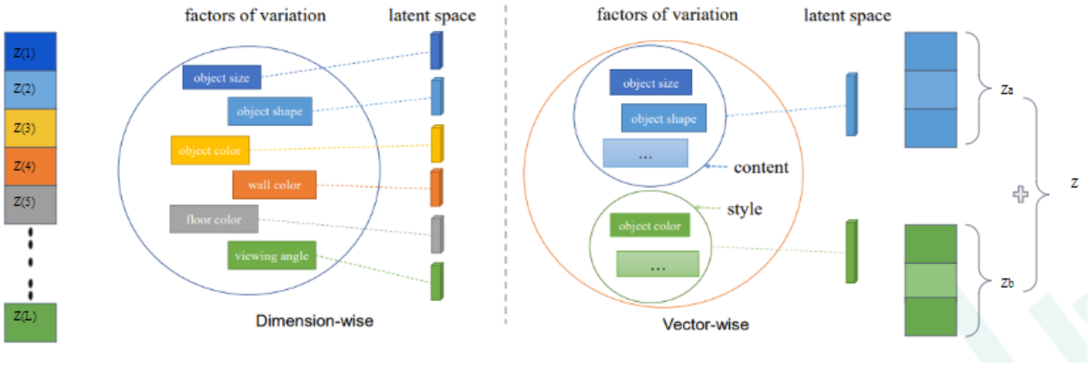


Figure 2.1: DRL classification based on the structure of representations of feature vector (Wang *et al.* (2022*b*))

Variational Autoencoders (VAEs) have been extensively utilized for disentanglement representation learning due to their ability to learn probabilistic latent variable models. The key idea behind VAE-based disentanglement methods is to enforce the latent space to be disentangled, meaning that different dimensions of the latent space correspond to independent and meaningful factors of variation in the data. An important variant of VAE for dimension-wise disentanglement is the β-VAE Higgins *et al.* (2017), which extends the traditional VAE by introducing a hyperparameter β to the loss function. The objective of β-VAE is to balance reconstruction accuracy and disentanglement of latent factors. The loss function for β-VAE is given by:

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \beta D_{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]$$

where $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is the approximate posterior, $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is the likelihood, D_{KL} denotes the KL divergence, and $p(\mathbf{z})$ is the prior on latent variables.

Another approach is FactorVAE Kim *et al.* (2019), which enhances disentanglement by introducing an additional independence constraint on the latent variables through a total correlation penalty term. The objective of FactorVAE is:

$$\mathcal{L}_{\text{FactorVAE}} = \mathcal{L}_{\text{rec}} + \gamma \text{TC}[\mathbf{z}] - \alpha D_{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})]$$

where $\mathcal{L}_{\text{rec}}$ is the reconstruction loss, TC$[\mathbf{z}]$ measures total correlation, and γ and α are hyperparameters controlling the balance between these terms.

Another popular approach of categorizing DRL methods is based on the learning method namely, supervised, unsupervised and weakly-supervised methods. VAE based methods are unsupervised methods. Well-known supervised and weakly-supervised methods are as follows:

- **Supervised Methods:** Bouchacourt *et al.* (2018) introduces the Multi-Level Variational Autoencoder (ML-VAE) that partitions samples into groups based on a selected factor f_s, where each group contains samples that share the same value of f_s. This approach is particularly useful in applications like image-to-image translation, where images within a group not only share the same label but also have the same posterior distribution of latent variables concerning f_s. This distribution depends on all samples in the group. Conversely, for factors other than f_s, the posterior distribution may vary for each individual sample. Eastwood and Williams (2018) demonstrate a structured approach to quantitatively assess the quality of disentangled representations when the ground-truth latent structure is known. They propose a method to objectively evaluate disentangled representations in machine learning models through three key criteria: disentanglement, completeness, and informativeness. The framework uses regression metrics to quantify how well each latent dimension captures a single ground-truth factor, how comprehensively all factors are represented, and the overall information preserved about the data.

- **Unsupervised Methods:** One notable approach in unsupervised disentanglement is based on VAEs. Traditional VAEs encode input data $\mathbf{x}$ into latent variables $\mathbf{z}$, with the objective to maximize the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})],$$

where $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ are the encoder and decoder distributions respectively, and D_{KL} denotes the KL divergence between the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the prior $p(\mathbf{z})$. While traditional VAEs aim for reconstruction and regularization, achieving disentanglement requires additional constraints or modifications. Recent advancements include β-VAE Ramakrishna *et al.* (2022), which introduces a hyperparameter β to balance between reconstruction accuracy and disentanglement of latent variables, as given earlier. In β-VAE the hyperparameter β controls the strength of the disentanglement regularization. Another approach is based on GANs, such as InfoGAN Chen *et al.* (2016), which extends the standard GAN framework to learn interpretable and disentangled representations by maximizing mutual information between latent variables and generated outputs. InfoGAN introduces an additional term to the GAN objective to explicitly encourage the learning of independent and interpretable factors.

- **Weakly Supervised Methods:** Weakly supervised disentanglement representation learning leverages limited and imprecise supervision to guide the learning of interpretable latent variables, balancing between the extremes of fully supervised and unsupervised methods. This approach is particularly valuable when full supervision is impractical or expensive to obtain. One critical insight into the challenges of unsupervised disentanglement is provided by Locatello *et al.* (2019) where they demonstrated that without inductive biases in the data or model, learning disentangled latent spaces in a purely unsupervised manner is theoretically impossible. Their work highlighted the necessity of some form of supervision, even if weak, to achieve meaningful disentanglement. In the work Chen and Batmanghelich (2020), the authors propose using pairwise similarities as a weak supervision signal. By leveraging pairs of data points that share some common factors of variation, the model is encouraged to learn latent representations where these shared factors are explicitly disentangled. In the work Rahiminasab *et al.* (2022), the authors tackle OoD reasoning by using weak supervision to disentangle latent factors that explain OoD behavior. For example, a model trained in urban environments with no precipitation will consider rural roads and rainy weather as OoD. Their approach involves logic-based variational autoencoders to reason about the sources of OoD. In SW-VAE Zhu *et al.* (2022), authors propose a novel approach where latent factor swapping is used as a weak supervision signal. If the latent representations are perfectly disentangled, swapping factors between data points should result in consistent reconstructions.

Training Strategy: There are two primary schemes for incorporating disentangled representations into downstream tasks: In joint optimization, the disentangled module and the task objective are optimized simultaneously. The task loss can guide the disentanglement process, resulting in representations that are both disentangled and informative for the task at hand. In the two-stage scheme, the disentangled module is trained independently, followed by applying the disentangled features in downstream tasks. In this case, the task loss cannot directly guide the disentanglement process, which might result in less optimal representations for specific tasks.

2.2.1 Loss Functions for Disentanglement

The design of loss functions is crucial for enforcing disentanglement and informativeness. Typically, the loss functions is designed as Wang *et al.* (2022*b*):

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{re}} + \lambda_2 \mathcal{L}_{\text{disen}} + \lambda_3 \mathcal{L}_{\text{task}},$$

where $\mathcal{L}_{\text{re}}$ denotes reconstruction loss, $\mathcal{L}_{\text{disen}}$ denotes disentanglement loss, and $\mathcal{L}_{\text{task}}$ denotes specific task loss.

1. **Reconstruction Loss** ($\mathcal{L}_{\text{re}}$): Ensures that the disentangled representation is semantically meaningful and can recover the original data.
2. **Disentanglement Loss** ($\mathcal{L}_{\text{disen}}$): Enforces the separation of different factors of variation within the representation.
3. **Task Loss** ($\mathcal{L}_{\text{task}}$): Directly related to the specific objective of the model, such as classification accuracy or regression error.

2.3 Vector-wise Disentanglement: Intra-class variance and invariance

The paper Lin *et al.* (2018) proposes a method to enhance fine-grained few-shot classification by disentangling and modeling intra-class variance and invariance. Given a dataset $\mathcal{X}$ consisting of N i.i.d images from M classes given as $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)})$, the data generation process involves an unobserved continuous random variable $\mathbf{z} = \mathbf{z}_V + \mathbf{z}_{Ik}$, representing the embedding features of the samples. The process includes three steps:

1. A value $\mathbf{z}_V^{(i)}$ is generated from some conditional distribution $p^*_{\boldsymbol{\theta}(z)}$, which represents the intra-class variance of sample i from class k.
2. $\mathbf{z}_{Ik}^{(i)}$ is the intra-class invariance of sample i from class k, and $\mathbf{z}^{(i)}$ is the sum of $\mathbf{z}_V^{(i)}$ and $\mathbf{z}_{Ik}^{(i)}$.
3. An image $\mathbf{x}^{(i)}$ is generated from some conditional distribution $p^*_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

Assuming that the prior $p^*_{\boldsymbol{\theta}(\mathbf{z})}$ and the likelihood $p^*_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})}$ are from parametric families $p_{\boldsymbol{\theta}}(\mathbf{z})$ and $p_{\boldsymbol{\theta}(\mathbf{x}|\mathbf{z})}$, the loss for modeling intra-class variance using a Monte Carlo estimator similar to VAE can be approximated as:

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \approx -\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_V|\mathbf{x}^{(i)}) \| p_{\boldsymbol{\theta}}(\mathbf{z}_V)) + \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}_V^{(i,l)})$$

Here, the prior distribution of $\mathbf{z}_V$ is assumed to be a centered isotropic multivariate Gaussian $p_{\boldsymbol{\theta}}(\mathbf{z}_V) = \mathcal{N}(\mathbf{z}_V; 0, \mathbf{I})$, and the approximate posterior is a multivariate Gaussian with a diagonal covariance:

$$q_{\boldsymbol{\phi}}(\mathbf{z}_V|\mathbf{x}^{(i)}) = \mathcal{N}(\mathbf{z}_V; \mu^{(i)}, \sigma^{2(i)}\mathbf{I})$$

The mean $\mu^{(i)}$ and standard deviation $\sigma^{(i)}$ of the posterior are approximated using the outputs of fully-connected layers. Using the re-parameterization trick, the objective function can be formulated as:

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathcal{X}_B) \approx{}& \frac{1}{2B}\sum_{i=1}^{B}\sum_{j=1}^{J}\left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right) \\ &+ \frac{1}{TB}\sum_{i=1}^{B}\sum_{t=1}^{L} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}_V^{(i,t)}) \end{aligned} \tag{2.1}$$

This can be denoted as $L_1 + L_2$, where L_1 enforces the distribution of intra-class variance to be isotropic centered Gaussian, and L_2 ensures that intra-class variance preserves sample-specific information.

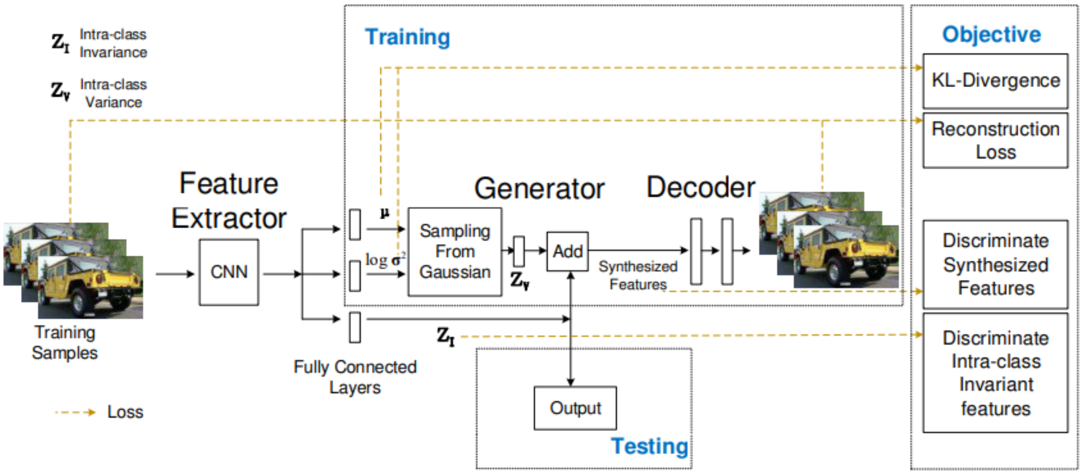


Figure 2.2: Deep Variational Metric Learning Framework (Lin *et al.* (2018)

Another paper Xu *et al.* (2021) focuses on disentangling class-specific features and handling intra-class variance to improve classification performance in few-shot learning scenarios where data is limited. The authors propose a framework that separates class-specific features from intra-class variations. This approach involves modeling the intra-class variations using a variational autoencoder (VAE), where the encoder learns to map input images to a latent space, capturing both inter-class and intra-class variations. These additional features are extracted and perform data augmentation with these new set of images, thereby imporoving the classification performance.

2.4 Domain Adaptation Methods for OoD Detection

Domain adaptation methods for OOD detection aim to bridge the gap between training and test distributions, making models more robust to unseen data. These methods include feature alignment Chen *et al.* (2019), domain-invariant representation learning Ganin and Lempitsky (2015), data augmentation Volpi *et al.* (2018) and transfer learning Kamath *et al.* (2019) based methods. By adapting the model to handle diverse and shifting data distributions, these techniques improve its ability to generalize and accurately detect OOD examples. In this work, we are particularly interested in disentangling into class-variant and class-invariant features, which we achieve using domain adaptation principle using the Gradient Reversal Layer (GRL).

2.4.1 GRL based Domain Adaptation

GRL has emerged as a pivotal component in domain adaptation and domain generalization tasks, serving to mitigate domain shift and promote domain-invariant representations. Introduced by Ganin and Lempitsky (2015), GRL operates by reversing the gradient flow during backpropagation, thereby encouraging the model to learn features that are insensitive to domain differences.

The GRL operates by multiplying the gradient of the domain classifier's loss with respect to the feature extractor's parameters by $-\lambda$, effectively reversing the direction of gradients during backpropagation:

$$\mathcal{L}_{total} = \mathcal{L}_{task} - \lambda \mathcal{L}_{domain}, \tag{2.2}$$

where $\mathcal{L}_{total}$ is the total loss function, and λ is a hyperparameter controlling the importance of domain alignment.The architecture consists of three main components: a feature extractor G_f, a label predictor G_y, and a domain discriminator G_d. We consider the functional:

$$E(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y, \boldsymbol{\theta}_d) = \sum_{i=1}^{N} \mathbf{1}_{d_i=0} L_y(G_y(G_f(\mathbf{x}_i; \boldsymbol{\theta}_f); \boldsymbol{\theta}_y), y_i) - \lambda \sum_{i=1}^{N} L_d(G_d(G_f(\mathbf{x}_i; \boldsymbol{\theta}_f); \boldsymbol{\theta}_d), y_i) \tag{2.3}$$

$$\begin{aligned} E(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y, \boldsymbol{\theta}_d) &= \sum_{\substack{i=1 \\ d_i=0}}^{N} L_y(G_y(G_f(\mathbf{x}_i; \boldsymbol{\theta}_f); \boldsymbol{\theta}_y), y_i) - \lambda \sum_{i=1}^{N} L_d(G_d(G_f(\mathbf{x}_i; \boldsymbol{\theta}_f); \boldsymbol{\theta}_d), y_i) \\ &= \sum_{\substack{i=1 \\ d_i=0}}^{N} L_y^i(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y) - \lambda \sum_{i=1}^{N} L_d^i(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d) \end{aligned} \tag{2.4}$$

Here, $L_y(\cdot, \cdot)$ is the loss for label prediction, $L_d(\cdot, \cdot)$ is the loss for the domain classification, while L_y^i and L_d^i denote the corresponding loss functions evaluated at the i-th training example. GRL seeks the parameters $\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_y, \hat{\boldsymbol{\theta}}_d$ that deliver a saddle point of the functional (2.4) is given as

$$(\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_y) = \arg\min_{\boldsymbol{\theta}_f, \boldsymbol{\theta}_y} E(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y, \hat{\boldsymbol{\theta}}_d) \tag{2.5}$$

$$\hat{\boldsymbol{\theta}}_d = \arg\max_{\boldsymbol{\theta}_d} E(\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_y, \boldsymbol{\theta}_d) \tag{2.6}$$

At the saddle point, the parameters $\boldsymbol{\theta}_d$ of the domain classifier minimize the domain classification loss (since it enters into (2.4) with the minus sign) while the parameters $\boldsymbol{\theta}_y$ of the label predictor minimize the label prediction loss. The feature mapping parameters $\boldsymbol{\theta}_f$ minimize the label prediction loss (i.e., the features are discriminative), while maximizing the domain classification loss (i.e., the features are domain-invariant). The parameter λ controls the trade-off between the two objectives that shape the features during learning. Below, we demonstrate that standard stochastic gradient solvers (SGD) can be adapted for the search of the saddle point (2.5),(2.6).

Optimization with Backpropagation

A saddle point of (2.5),(2.6) can be found as a stationary point of the following stochastic updates:

$$\boldsymbol{\theta}_f \leftarrow \boldsymbol{\theta}_f - \mu \left(\frac{\partial L_i^y}{\partial \boldsymbol{\theta}_f} - \lambda \frac{\partial L_i^d}{\partial \boldsymbol{\theta}_f} \right) \tag{2.7}$$

$$\boldsymbol{\theta}_y \leftarrow \boldsymbol{\theta}_y - \mu \frac{\partial L_i^y}{\partial \boldsymbol{\theta}_y} \tag{2.8}$$

$$\boldsymbol{\theta}_d \leftarrow \boldsymbol{\theta}_d - \mu \frac{\partial L_i^d}{\partial \boldsymbol{\theta}_d} \tag{2.9}$$

where μ is the learning rate (which can vary over time). The updates in (2.7),(2.8),(2.9) are very similar to stochastic gradient descent (SGD) updates for a feed-forward deep model that comprises feature extractor fed into the label predictor and into the domain classifier. The difference is the $-\lambda$ factor in (2.7): the difference is important, as without such factor, stochastic gradient descent would try to make features dissimilar across domains in order to minimize the domain classification loss. Although direct implementation of (2.7)-(2.9) as SGD is not possible, it is highly desirable to reduce them to some form of SGD.

Such reduction can be accomplished by introducing a special GRL defined as follows. The GRL has no parameters associated with it (apart from the meta-parameter λ, which is not updated by backpropagation). During the forward propagation, GRL acts as an identity transform. During the backpropagation though, GRL takes the gradient from the subsequent level, multiplies it by $-\lambda$ and passes it to the preceding layer. As the backpropagation process passes through the GRL, the partial derivatives of the loss that is downstream of the GRL (i.e., L_d) w.r.t. the layer parameters that are upstream of the GRL (i.e., $\boldsymbol{\theta}_f$) get multiplied by $-\lambda$, i.e., $\frac{\partial L_d}{\partial \boldsymbol{\theta}_f}$ is effectively replaced with $-\lambda \frac{\partial L_d}{\partial \boldsymbol{\theta}_f}$.

Therefore, running SGD in the resulting model implements the updates (2.7)-(2.9) and converges to a saddle point of (2.4). Mathematically, we can formally treat the gradient reversal layer as a "pseudo-function" $R_\lambda(\mathbf{x})$ defined by two (incompatible) equations describing its forward- and back propagation behavior:

$$R_\lambda(\mathbf{x}) = \mathbf{x}, \qquad \frac{dR_\lambda}{d\mathbf{x}} = -\lambda \mathbf{I} \tag{2.10}$$

where $\mathbf{I}$ is an identity matrix. We can then define the objective "pseudo-function" of $(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y, \boldsymbol{\theta}_d)$ that is being optimized by the stochastic gradient descent:

$$\begin{aligned}\tilde{E}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_y, \boldsymbol{\theta}_d) = &\sum_{i=1}^{N} \mathbf{1}_{d_i=0} L_y(G_y(G_f(\mathbf{x}_i; \boldsymbol{\theta}_f); \boldsymbol{\theta}_y), y_i) \\ &+ \sum_{i=1}^{N} L_d(G_d(R_\lambda(G_f(\mathbf{x}_i; \boldsymbol{\theta}_f)); \boldsymbol{\theta}_d), y_i)\end{aligned} \tag{2.11}$$

Running updates (2.7)-(2.9)can then be implemented as doing SGD for (2.11) and leads to the emergence of features that are domain-invariant and discriminative at the same time. After the learning, the label predictor $y(x) = G_y(G_f(x; \boldsymbol{\theta}_f); \boldsymbol{\theta}_y)$ can be used to predict labels for samples from the target domain (as well as from the source domain).

CHAPTER 3

Proposed Method: NORD-F

In this work, we address the challenge of detecting near-OOD samples through disentanglement representation learning where we extract class variant and class invariant features in fine-grained datasets with a goal to improve near OoD detection. Towards this, we introduce a novel approach which we refer to as NORD-F, for near-OOD detection in fine-grained environment. The method is based on vector-wise disentanglement and domain-adaptation based GRL application. We provide an overview of the proposed NORD-F framework in Fig. 3.1.

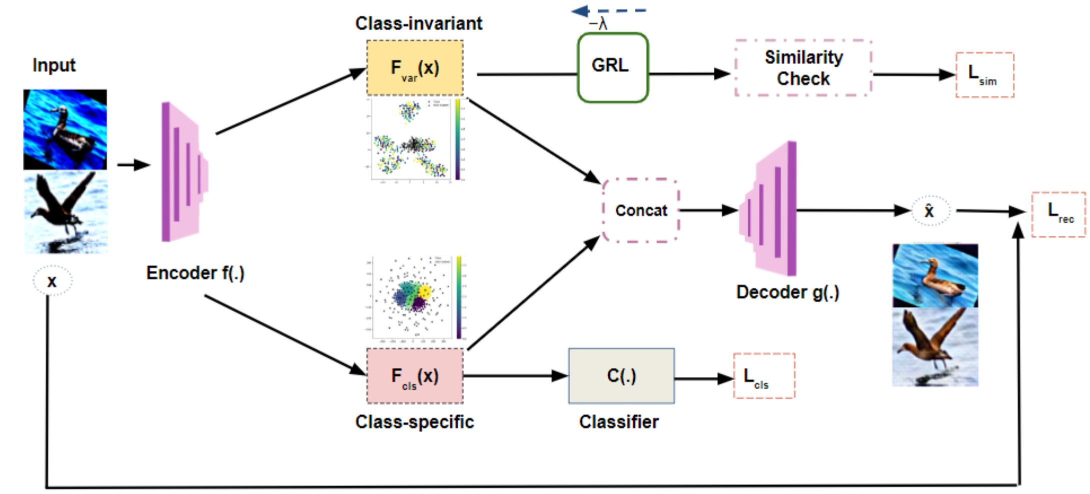


Figure 3.1: The figure depicts the three branches, namely the variation, classifier, and reconstruction branches. The loss functions used are given as $\mathcal{L}_{\text{sim}}$, $\mathcal{L}_{\text{rec}}$, and $\mathcal{L}_{\text{cls}}$. The variation branch focuses on the class-invariant features, and the classifier branch focuses on the class-specific features, as depicted in the TSNE. The reconstructed images are obtained at the output of the decoder as shown.

3.0.1 Branch-Wise Description

Variation Branch: The role of the variation branch is to encode the class-irrelevant information of image samples. This is achieved by incorporating a GRL in such a way that the effective features F_{var} emphasize the differences.

Consider a pair of images $\{\mathbf{x}_i, \mathbf{x}_j\}$. The variation branch is characterized by the GRL alongside the similarity block which, together measure the variation feature similarity between any two samples. The GRL acts as an identity transform in forward pass, and it multiples the gradient from the subsequent level by a constant $-\lambda$ during back-propagation. During training, we construct positive and negative pairs from the sample batch using the labels. The similarity module outputs a score $s_{i,j} \in [0, 1]$ indicating the probability that the pair $\{\mathbf{x}_i, \mathbf{x}_j\}$ are from the same class as $s_{i,j} = c_\psi(F_{var}(\mathbf{x}_i), F_{var}(\mathbf{x}_j))$, where c_ψ is the cosine distance between $F_{var}(\mathbf{x}_i)$ and $F_{var}(\mathbf{x}_j)$.

Classifier Branch: In this branch, we focus on extracting features that are discriminative in nature, such that it helps classifying between classes. Given an image $\mathbf{x}_i$, we extract their class-specific feature representations F_{cls} using a neural network encoder $f(\cdots)$ and classifier module $C(\cdots)$ Hence the predicted output for this sample is obtained as

$$\hat{y}_i = C(f(F_{cls}(\mathbf{x}_i))). \tag{3.1}$$

Reconstruction Branch: To preserve the image information and achieve feature disentanglement, a decoder module with a MLP module and a decoder network $g(\cdot)$ is used, which combines the classification and variation branches for image reconstruction. The features isolated by the variation and the classification branch are together input into the decoder, which leads to the image reconstruction given as

$$\hat{\mathbf{x}}_i = g(F_{cls}, F_{var}). \tag{3.2}$$

3.0.2 Loss Functions

The objective function consists of the the similarity loss L_{sim} for the variation branch, cross-entropy loss L_{cls} for the classification task, and a reconstruction loss L_{rec}.

To remove the class-specific information in the variation branch, binary cross entropy loss is used to optimize the variation feature maps. The similarity loss is the binary cross-entropy loss between the similarity score $s_{i,j} \in [0, 1]$ and a label $l \in \{0, 1\}$, indicating whether the two images are from the same class:

$$\mathcal{L}_{\text{sim}} = -\sum_{i,j \in \mathcal{P}} \left(l_{i,j} \log(s_{i,j}) + (1 - l_{i,j}) \log(1 - s_{i,j}) \right), \tag{3.3}$$

where $(i, j) \in \mathcal{P}$ training pairs represents the pairs for which the similarity is calculated using cosine distance and $l_{i,j} = 0$ or 1 indicates the ground truth whether the two images are same or different. We minimize L_{sim} in training, and apply GRL to reverse the gradient during back-propagation to achieve feature disentangling and minimize the class-specific information captured by the variation branch.

To preserve class-related features, we minimize the cross-entropy loss L_{cls} for the classification branch. Simultaneously, the features $F_{cls}(\mathbf{x})$ are fed into a classifier to predict their classes. The classification loss is given by the cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\sum_{i} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)), \tag{3.4}$$

where y_i are the ground truth label for x_i.

To ensure that the disentangled discriminative and variation features can jointly restore the original input image, perceptual loss is applied after decoding for reconstruction. For an input x_i, the reconstructed image is represented as $\hat{x}_i$. The decoder is represented as decoder ($g(\cdot)$). We use a perceptual loss based on feature maps ϕ from a pre-trained VGG network to compare the reconstructed and original images:

$$\mathcal{L}_{\text{perc}} = \sum_{i} \|\phi(\hat{\mathbf{x}}_i) - \phi(\mathbf{x}_i)\|_2^2 \,. \tag{3.5}$$

Gradients of loss components due to GRL The gradients of the total loss with respect to the model parameters θ is computed as follows:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \theta} = -\lambda_{\text{sim}} \frac{\partial \mathcal{L}_{\text{sim}}}{\partial \theta}, \lambda_{\text{cls}} \frac{\partial \mathcal{L}_{\text{cls}}}{\partial \theta}, \lambda_{\text{perc}} \frac{\partial \mathcal{L}_{\text{perc}}}{\partial \theta}, \tag{3.6}$$

Only the Similarity branch loss component will be reversed due to GRL with the strength of λ.

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, we evaluate the effectiveness of the proposed NORD-F framework in detecting near OoD samples in fine-grained datasets. The focus is to illustrate its capability in enhancing classification robustness and identifying near-ood samples. By disentangling relevant features from irrelevant ones, our method aims to improve the model's ability to distinguish subtle visual differences crucial for fine-grained classification tasks. In particular, we demonstrate the following (a) as compared to OOD-aware baselines, NORD-F performs better in terms of the OOD metrics, (b) TSNE visualization clearly illustrates that the proposed technique disentangles the feature representation as class-invariant and class-specific features. In the sequel, we describe the datasets, architecture, baselines and the metrics used in this work, followed by the experimental results.

4.1 Datasets

Fine-grained image classification datasets often contain images from categories that are visually very similar, such as different species of birds, breeds of dogs, or models of aircraft. Such datasets are used to evaluate the capability of models to learn and generalize fine-grained visual distinctions. The following fine-grained datasets were used for our experiments:

- **CUB-200-2011 (Caltech-UCSD Birds-200-2011)**: A widely used dataset containing images of 200 bird species. It includes 11,788 images, with each species having roughly 30 training and 20 testing images. The dataset also provides annotations for bounding boxes, part locations, and attributes, which can be useful for various fine-grained classification tasks Wah *et al.* (2011).

- **Stanford Dogs**: A dataset containing images of 120 breeds of dogs from around the world. It has a total of 20,580 images, with each class having approximately 150 images. This dataset is useful for fine-grained classification due to the significant visual similarity between different breeds Khosla *et al.* (2011).

- **FGVC-Aircraft (Aircraft Variant Recognition)**: A dataset designed for fine-grained aircraft classification. It includes 10,000 images of aircraft, divided into 100 different variants. The images in this dataset are annotated with detailed information about aircraft models, making it a challenging fine-grained classification task Maji *et al.* (2013).

Table 4.1: Details of Fine-Grained Datasets Used in Experiments

Dataset	Total Images	Number of Classes	Annotations
CUB-200-2011	11,788	200	Bounding boxes, part locations, attributes
Stanford Dogs	20,580	120	Bounding boxes
FGVC-Aircraft	10,000	100	Variant annotations

4.1.1 Dataset Splits for Experiments

For the purpose of our experiments, we created specific splits of the datasets to evaluate our model's performance on in-distribution (ID) and near out-of-distribution (OOD) data. The splits were defined as follows:

- **In-Distribution (ID) Dataset**:With total number of N classes, we selected the top k classes with the most number of images as the ID dataset. These classes were used for training, validation, and testing.

- **Near Out-of-Distribution (OOD) Dataset**: The remaining other N-k classes were considered as the near OOD dataset. This setup ensures that the model encounters categories that are similar but not identical to the training classes, making the task more challenging.

- **Far Out-of-Distribution (OOD) Dataset**: A completely different dataset was used as the far OOD dataset. For example, if the CUB-200-2011 bird dataset was used for training, the Oxford Flowers dataset Nilsback and Zisserman (2008) was used as the far OOD dataset. This setup evaluates the model's ability to handle entirely different categories .

- **Training, Validation, and Testing Splits**: The training set consisted of images from the top k classes used for training. The validation and testing sets were created from the unseen images of the same k classes to ensure that the model was evaluated on new, unseen data within the same distribution (Train-Test-VAl : 80% - 10% - 10%).

In our experiments, we compared the performance of the model on the ID dataset (training, validation, and testing splits) with its performance on the near and far OOD datasets. This comparison allowed us to evaluate the model's ability to generalize to new, unseen categories that are visually similar (near OOD) and completely different (far OOD) from the training classes.

4.2 Architecture Description

Our model architecture comprises three main branches designed to handle fine-grained classification tasks and disentanglement of image features. The model architecture consists of multiple components including a backbone network, which takes two images as input along with their corresponding targets. Additionally, a variable l is provided as input, indicating whether the two images share the same targets ($l = 0$) or have different targets ($l = 1$)). The architecture also consists of GRL, convolutional blocks, classifier, similarity block (c_ψ), and a decoder ($g(\cdot)$) in the reconstruction branch. Below is a detailed description of each component:

- **Backbone Network**
  - Uses a pre-trained DenseNet-121 model.
  - The final fully connected layer is removed.
  - Features extracted from this network are used as input for other parts of the model.
- **Gradient Reversal Layer**
  - Custom function implemention of GRL.
  - Multiplies the gradient by a negative scalar during the backward pass, effectively reversing the gradient with the weight λ.
- **Convolutional Block**
  - Consists of a convolutional layer, batch normalization, and ReLU activation.
- **Classifier**
  - Consists of three fully connected layers.
  - Uses ReLU activations and dropout for regularization.
- **Variation Branch**
  - Includes a convolutional block.
  - Computes the cosine similarity between feature vectors.
  - Uses gradient reversal on the input features before applying the convolutional block.
- **Classifier Branch**
  - Includes a convolutional block followed by a classifier.
- **Reconstruction Branch**
  - Includes a decoder that reconstructs the input image from concatenated feature maps.
  - The decoder is composed of several upsample blocks, each containing a ConvTranspose2d layer, batch normalization, ReLU activation, and two residual blocks.

4.2.1 Evaluation Metrics

To evaluate the performance of our model, we employ several metrics commonly used in Out-of-Distribution (OoD) detection tasks. These metrics provide a comprehensive assessment of the model's ability to distinguish between in-distribution (ID) and out-of-distribution (OOD) samples.

- **Area Under the Receiver Operating Characteristic Curve (AUROC)**
  - Measures the model's ability to discriminate between positive and negative classes.4.2
- **Area Under the Precision-Recall Curve (AUPR)**4.2
  - **AUPR-in**: Evaluates the precision and recall for ID samples.
  - **AUPR-out**: Assesses the precision and recall for OoD samples.
- **Entropy Curves** 4.5
  - Graphically assess the entropy values of the model's predictions for ID and OoD samples.
- **Entropy Receiver Operating Characteristic Curves (Entropy ROCs)** 4.7
  - Evaluate the ROC curves based on entropy values to distinguish ID from OoD samples. This is a novel metric introduced in this thesis for better evaluation of OoD detection methodologies.
- **Confidence Curves** 4.6
  - Analyze the confidence scores of the model for ID and OoD samples to differentiate between them.

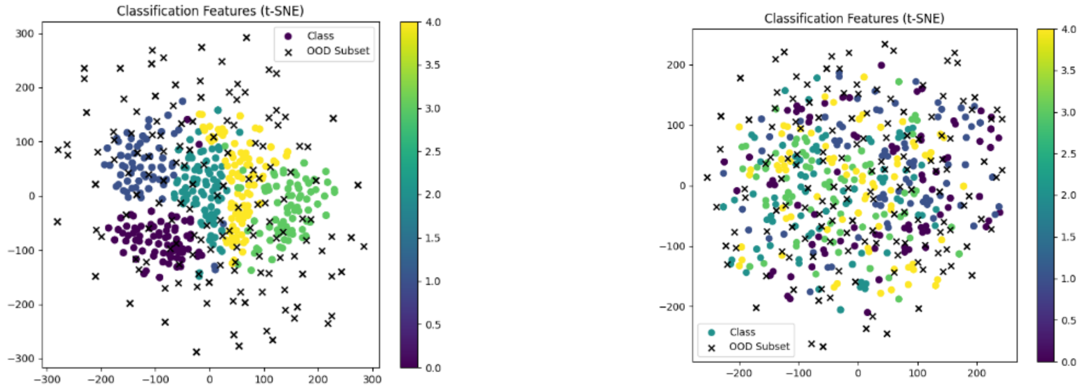
4.2.2 Baselines for Comparison

To evaluate the effectiveness of our proposed method, we compare it against several established baselines in the field of OoD detection. These baselines provide a benchmark for assessing the performance improvements achieved by our approach. Described in 2.1

- **Energy-Based OoD Detection**: Leverages the energy score derived from the model's output logits.
- **Temperature Scaling**: A post-processing technique that adjusts the confidence scores by scaling the logits.
- **Logit Norm**: Normalizes the logits to enhance the discriminative power of the model.

4.3 Ablation Study With Experimental Results

Variation branch's GRL values: For GRL branch, we experimented with multiple values of λ along with different GRL annealing techniques. Typically, the annealing strategy involves a schedule where the magnitude of the gradient reversal starts small and increases over time. This can be done linearly or with a more complex schedule depending on the specific requirements of the task. Through our experiments we discovered that applying small amount of constant λ values after certain epochs is more beneficial than constantly increasing it. As demonstrated in 4.1.



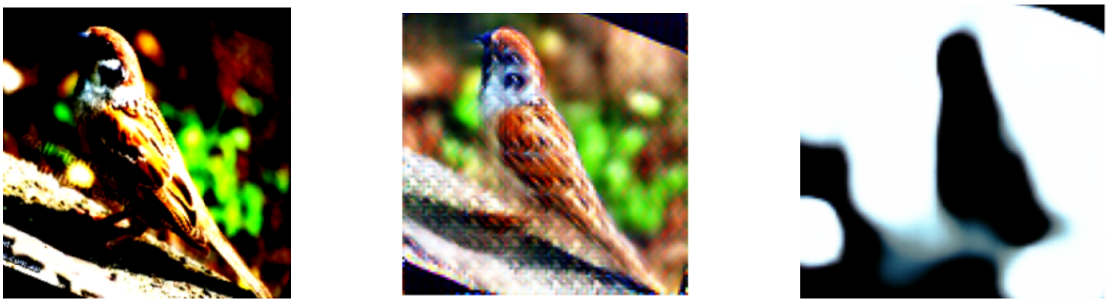
(a) With 0.1 constant λ value

(b) With 0.5 linearly annealed

Figure 4.1: Classification branch's feature plots at different λ values

Figure 4.8a shows more class-wise clusters with constant λ values as compared to 4.8b which shows dispersed features.

Reconstruction Loss: For reconstruction purpose we tried with MSE loss and perceptual loss, results show that perceptual loss gives better reconstruction than MSE loss as shown in 4.2



(a) Original Image

(b) Perceptual loss

(c) MSE loss

Figure 4.2: Reconstruction comparison with different loss functions

Training Strategy (Joint optimization V/S individual optimization of loss(es)

During training, each branch of the model optimizes its respective loss function individually. The variation branch minimizes the BCE loss based on cosine distance, the classification branch minimizes CE loss, and the reconstruction branch minimizes perceptual loss. We also experimented with a combined training approach where all losses are back-propagated together using a single optimizer, aiming to optimize the overall performance of the model across all tasks simultaneously. We used Adam optimizer with learning rate of 0.001 for each branch. We tried with the learning rate of 0.0001 as well but we found loss convergence issue with it during training time.

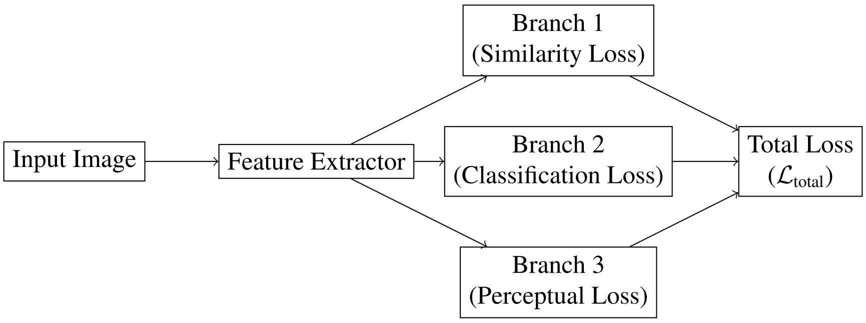


Figure 4.3: Joint Optimization through Total Loss

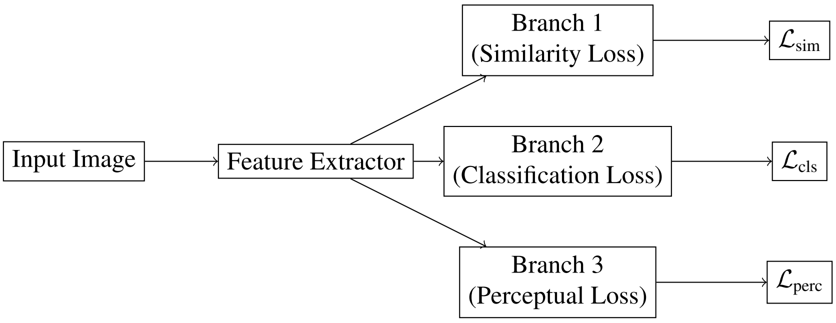


Figure 4.4: Individual Loss Optimization

Vanilla Classifier V/s NORD-F Comparison: We compare confidence curve, entropy curve and ROC curve(entropy based) obtained from vanilla classifier as well as NORD-F (which is vanilla classifier + additonal novel branches).

Figure 4.5 depicts that there is a better separation between ID and near-OoD entropies. With our framework, the entropy peaks get down showing lesser confidence for OoD data.

Figure 4.7 shows that with our framework, ROC shows better results, meaning that our framework better separates IDs and OoDs

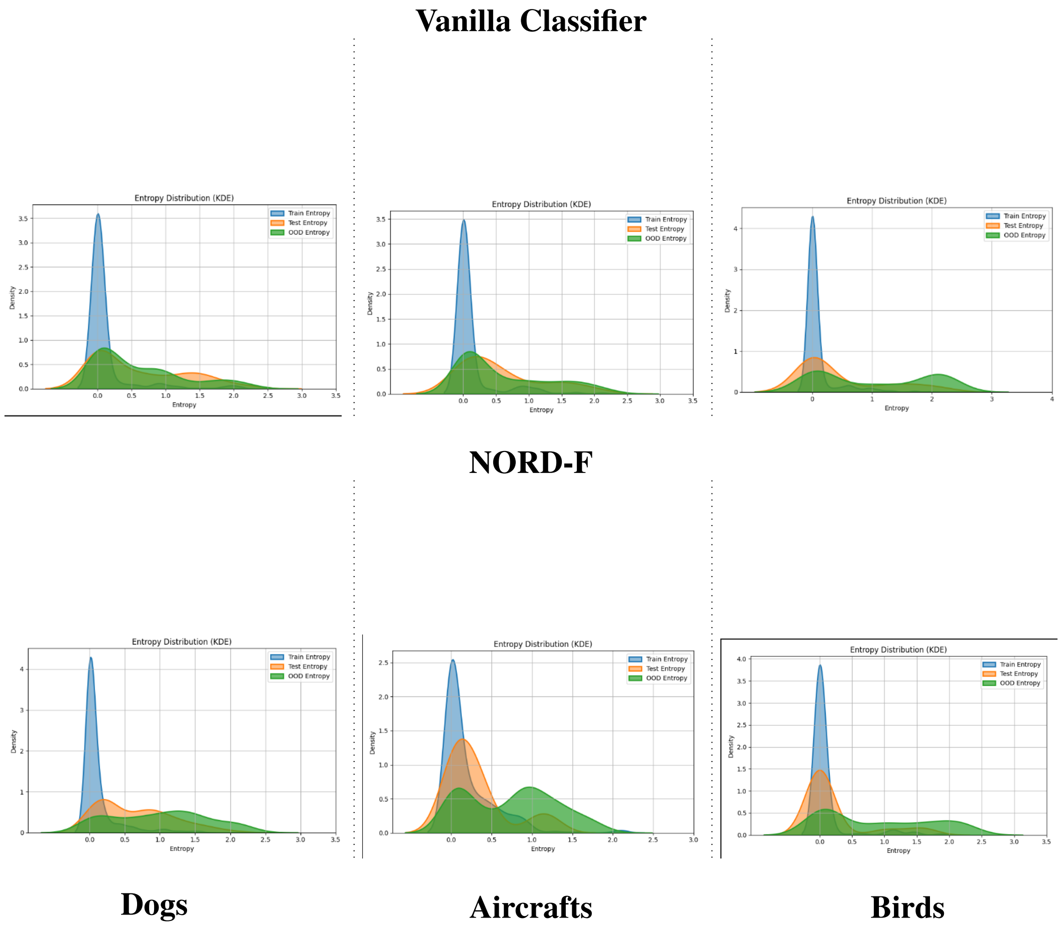
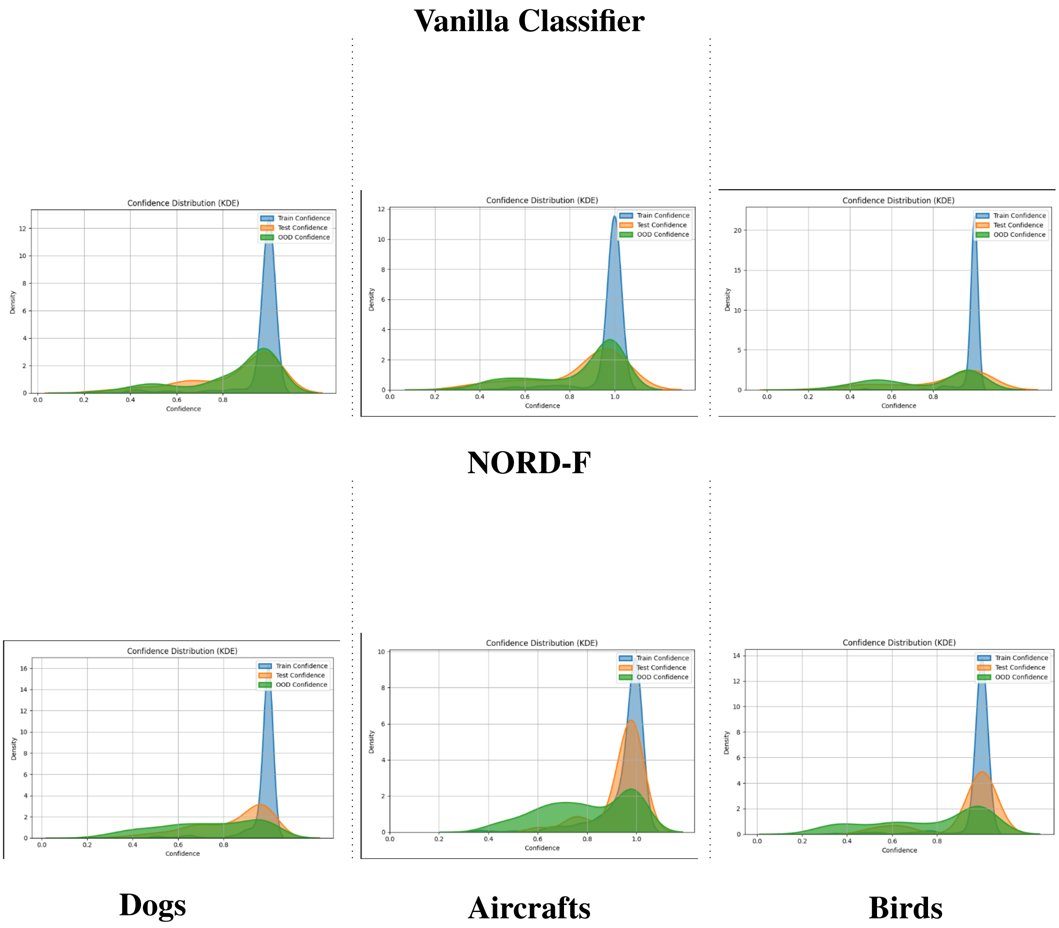


Figure 4.5: Entropy comparison for a vanilla classifier with NORD-F



Vanilla Classifier

NORD-F

Dogs

Aircrafts

Birds

Figure 4.6: Confidence curve comparison for a vanilla classifier with NORD-F

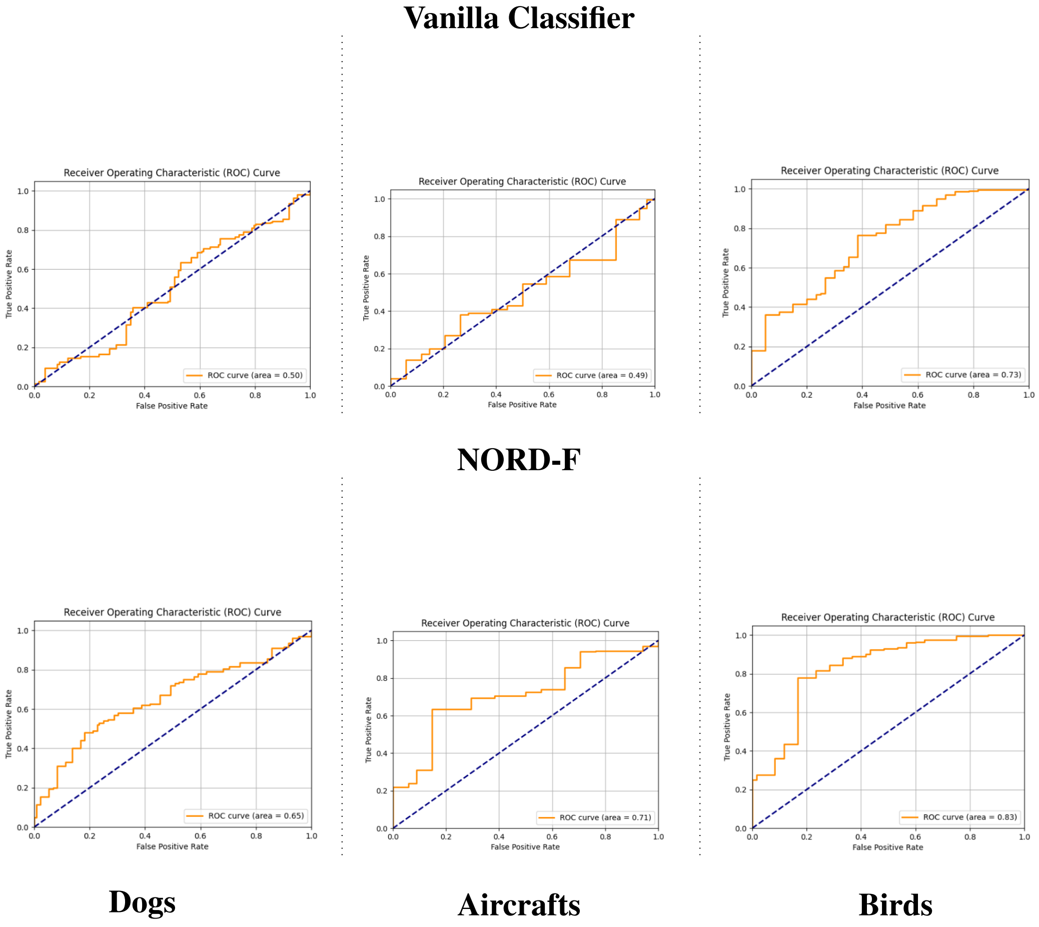
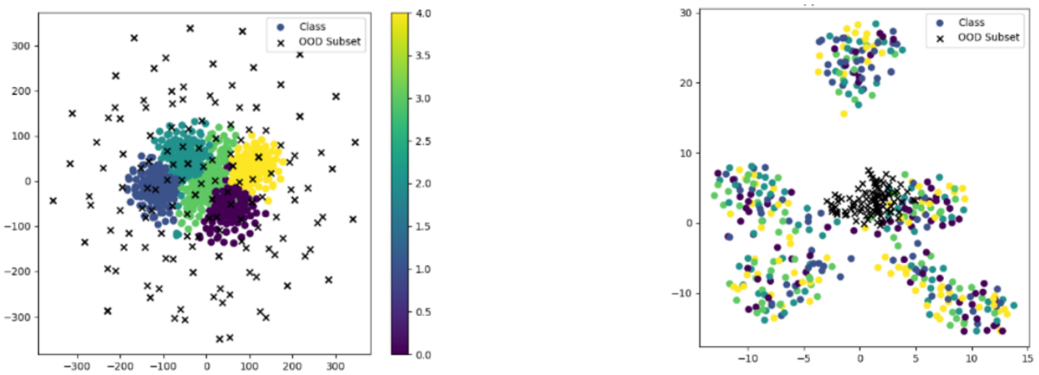


Figure 4.7: ROC comparison for a vanilla classifier with NORD-F

4.3.1 Evaluations From Different Branches

Feature plots from variation branch F_{var} and classifier branch F_{cls} 4.8



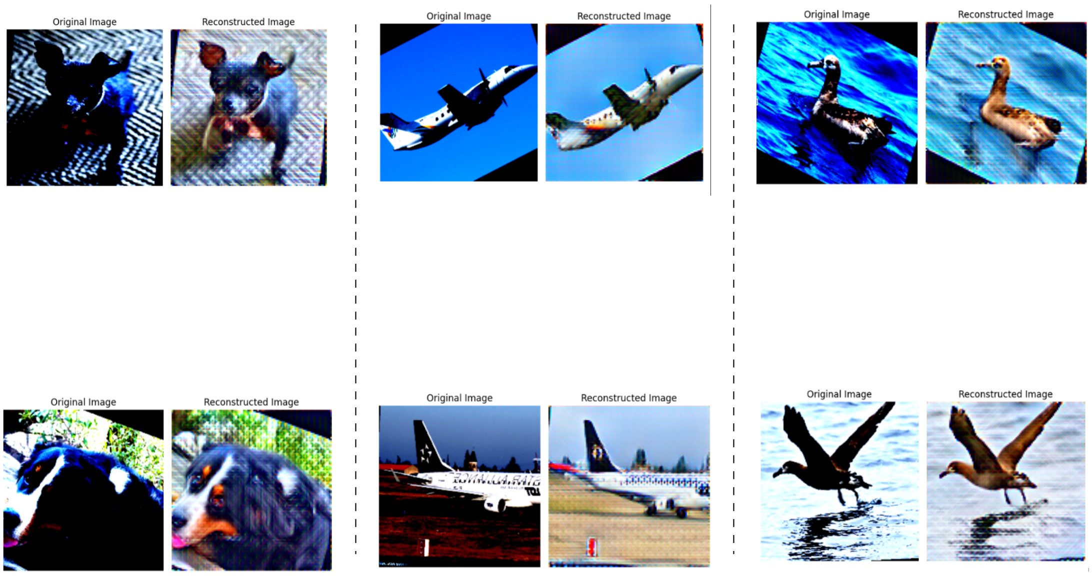
(a) Features from classifier branch

(b) Features from variation branch

Figure 4.8: Feature plots from different branches (this is from Dog dataset)

Figure 4.8a shows class-wise clusters while 4.8b shows features which are more common across all the classes. It also shows that OoD class is more concentrated with common features (similarity branch) showing that it contains more irrelevant features.

Reconstructed images for each dataset used:4.9



(a) Dogs

(b) Aircrafts

(c) Birds

Figure 4.9: Reconstruction observed while training with three different datasets using perceptual loss

4.4 Comaprison with Baselines: Evaluation Metrics

Using the evaluation metrics provided earlier, we comprehensively evaluate the effectiveness of our model in distinguishing between ID and OoD samples, ensuring robust performance in real-world scenarios.

Below are the results obtained from the experiments (For Baseline description refer section 2.1)

Table 4.2: Comparison with OoD Detection Baseline Methods

Din	**Method**	**AUROC (Near/Far)**	**AUPR-IN(Near/Far)**	**AUPR-OUT(Near/Far)**
Dogs	TempScale	0.7858 / 0.7421	0.7858/ 0.7421	0.4248 / 0.3344
	Energy	0.1942 / 0.2179	0.7127 / 0.7377	0.0965 / 0.0991
	LogitNorm	0.5337 / 0.5487	0.9217 / 0.9238	0.3287 / 0.3691
	NORD-F(ours)	**0.7954 / 0.8398**	**0.9515 / 0.9660**	0.4226 /**0.4324**
Aircraft	TempScale	0.8378 / 0.8708	0.9431 / 0.9610	0.5680 / 0.6082
	Energy	0.1536 / 0.0903	0.5920 / 0.5700	0.1415 / 0.1364
	LogitNorm	0.5403 / 0.5552	0.8868 / 0.8897	0.5279 / 0.5592
	NORD-F(ours)	**0.8531** / 0.7811	**0.9463 / 0.9083**	**0.6394** / 0.5343
Birds	TempScale	0.7816 / 0.8756	0.9191 / 0.9561	0.5909 / 0.6791
	Energy	0.1670 / 0.1070	0.5809 / 0.5621	0.1497 / 0.1440
	LogitNorm	0.6047 / 0.5590	0.8947 / 0.8855	0.6355 / 0.5708
	NORD-F(ours)	**0.8072** / 0.6767	**0.9243** / 0.8508	0.5122 / 0.4098

In the table 4.2, the numbers in bold represent the cases where our model works better than other baseline methods.

CHAPTER 5

CONCLUSION

In this thesis, we have explored various aspects of DRL and its application to OoD detection. In this thesis, we tackled the challenging problem of near-OoD detection using fine-grained datasets. Our approach focused on disentangling features into relevant and irrelevant parts, thereby improving the robustness of OoD detection. By evaluating various baseline methods with metrics such as AUROC, AUPR, entropy plots, and confidence scores, we demonstrated the effectiveness of our disentanglement-based approach. Our methods turns out to be superior since we are not using toy datasets like MNIST, SVHN, CIFAR 10, etc. for comparison of our methodology. Also, our method make use of zero outlier exposure during training considering the real life settings since we can never predict what sort of outliers can come up during inference time. The results show that disentangled representations significantly enhance the ability to detect OoD samples, particularly in fine-grained settings, paving the way for more reliable and interpretable models in critical applications.

CHAPTER 6

FUTURE WORK

While significant progress has been made in DRL and OoD detection, numerous avenues remain for future research. Despite the promise shown by current OoD detection methods, their robustness against adversarial attacks remains a pressing concern. Future efforts should focus on developing adversarially robust models that maintain high performance even in the presence of malicious perturbations. Additionally, while this work primarily concentrated on visual datasets, the principles of DRL and OoD detection are applicable across various domains, including natural language processing, audio signal processing, and medical imaging. Exploring these applications could reveal new challenges and opportunities. Furthermore, integrating ViT-based approaches with other disentanglement methods offers a promising direction to enhance performance and generalization capabilities. By addressing these areas, future research can continue to advance the field and expand the applicability of these techniques.

REFERENCES

1. **Bitterwolf, J.**, **A. Meinke**, and **M. Hein** (2020). Certifiably adversarially robust detection of out-of-distribution data. *Advances in Neural Information Processing Systems*, **33**, 16085–16095.

2. **Bouchacourt, D.**, **R. Tomioka**, and **S. Nowozin**, Multi-level variational autoencoder: Learning disentangled representations from grouped observations. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. 2018.

3. **Chen, C.**, **W. Xie**, **W. Huang**, **Y. Rong**, **X. Ding**, **Y. Huang**, **T. Xu**, and **J. Huang**, Progressive feature alignment for unsupervised domain adaptation. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

4. **Chen, J.** and **K. Batmanghelich**, Weakly supervised disentanglement by pairwise similarities. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34. 2020.

5. **Chen, J.**, **Y. Li**, **X. Wu**, **Y. Liang**, and **S. Jha**, Atom: Robustifying out-of-distribution detection using outlier mining. *In Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*. Springer, 2021.

6. **Chen, X.**, **Y. Duan**, **R. Houthooft**, **J. Schulman**, **I. Sutskever**, and **P. Abbeel** (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, **29**.

7. **Choi, J.**, **C. Yoon**, **J. Bae**, and **M. Kang** (2021). Robust out-of-distribution detection on deep probabilistic generative models. *arXiv preprint arXiv:2106.07903*.

8. **Daxberger, E.** and **J. M. Hernández-Lobato** (2019). Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*.

9. **Denton, E. L.** *et al.* (2017). Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems*, **30**.

10. **Du, X.**, **Z. Wang**, **M. Cai**, and **Y. Li** (2022). Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*.

11. **Eastwood, C.** and **C. K. Williams**, A framework for the quantitative evaluation of disentangled representations. *In 6th International Conference on Learning Representations*. 2018.

12. **Feng, Y.**, **D. J. X. Ng**, and **A. Easwaran** (2021). Improving variational autoencoder based out-of-distribution detection for embedded real-time applications. *ACM Transactions on Embedded Computing Systems (TECS)*, **20**(5s), 1–26.

13. **Floto, G.**, **S. Kremer**, and **M. Nica**, The tilted variational autoencoder: Improving out-of-distribution detection. *In The Eleventh International Conference on Learning Representations*. 2023.

14. **Fort, S.**, **J. Ren**, and **B. Lakshminarayanan** (2021). Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, **34**, 7068–7081.

15. **Ganin, Y.** and **V. Lempitsky**, Unsupervised domain adaptation by backpropagation. *In International conference on machine learning*. PMLR, 2015.

16. **Guo, C.**, **G. Pleiss**, **Y. Sun**, and **K. Q. Weinberger**, On calibration of modern neural networks. *In Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

17. **Hendrycks, D.** and **K. Gimpel** (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

18. **Hendrycks, D.**, **N. Mu**, **E. D. Cubuk**, **B. Zoph**, **J. Gilmer**, and **B. Lakshminarayanan** (2019). Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*.

19. **Higgins, I.**, **L. Matthey**, **A. Pal**, **C. P. Burgess**, **X. Glorot**, **M. M. Botvinick**, **S. Mohamed**, and **A. Lerchner** (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, **3**.

20. **Jiang, W.**, **Y. Ge**, **H. Cheng**, **M. Chen**, **S. Feng**, and **C. Wang**, Read: Aggregating reconstruction error into out-of-distribution detection. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37. 2023.

21. **Kamath, U.**, **J. Liu**, **J. Whitaker**, **U. Kamath**, **J. Liu**, and **J. Whitaker** (2019). Transfer learning: Domain adaptation. *Deep learning for NLP and speech recognition*, 495–535.

22. **Khosla, A.**, **N. Jayadevaprakash**, **B. Yao**, and **L. Fei-Fei** (2011). Stanford dogs dataset. *Stanford University*. URL `http://vision.stanford.edu/aditya86/ImageNetDogs/`.

23. **Kim, M.**, **Y. Wang**, **P. Sahu**, and **V. Pavlovic** (2019). Relevance factor vae: Learning and identifying disentangled factors. *arXiv preprint arXiv:1902.01568*.

24. **Lee, K.**, **K. Lee**, **H. Lee**, and **J. Shin** (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, **31**.

25. **Lee, S.**, **S. Cho**, and **S. Im**, Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.

26. **Liang, S.**, **Y. Li**, and **R. Srikant**, Enhancing the reliability of out-of-distribution image detection in neural networks. *In International Conference on Learning Representations*. 2018.

27. **Lin, X.**, **Y. Duan**, **Q. Dong**, **J. Lu**, and **J. Zhou**, Deep variational metric learning. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

28. **Liu, J.**, **Z. Shen**, **Y. He**, **X. Zhang**, **R. Xu**, **H. Yu**, and **P. Cui** (2021*a*). Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*.

29. **Liu, Q.**, **Z. Wu**, and **D. Tao** (2021*b*). Deep dive into semi-supervised elbo for improving classification performance. *arXiv preprint arXiv:2108.12734*.

30. **Liu, W.**, **X. Wang**, **J. Owens**, and **Y. Li** (2020). Energy-based out-of-distribution detection. *Advances in neural information processing systems*, **33**, 21464–21475.

31. **Locatello, F.**, **S. Bauer**, **M. Lucic**, **G. Raetsch**, **S. Gelly**, **B. Schölkopf**, and **O. Bachem**, Challenging common assumptions in the unsupervised learning of disentangled representations. *In international conference on machine learning*. PMLR, 2019.

32. **Maji, S.**, **E. Rahtu**, **J. Kannala**, **M. Blaschko**, and **A. Vedaldi**, Fine-grained visual classification of aircraft. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013. URL `http://www.robots.ox.ac.uk/~vgg/data/fgvc-aircraft/`.

33. **Mehta, D.**, **Y. Gal**, **A. Bowling**, **P. Bonnington**, and **Z. Ge**, Out-of-distribution detection for long-tailed and fine-grained skin lesion images. *In International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022.

34. **Mukhoti, J.**, **T.-Y. Lin**, **B.-C. Chen**, **A. Shah**, **P. H. Torr**, **P. K. Dokania**, and **S.-N. Lim**, Raising the bar on the evaluation of out-of-distribution detection. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

35. **Nguyen, A. T.**, **F. Lu**, **G. L. Munoz**, **E. Raff**, **C. Nicholas**, and **J. Holt**, Out of distribution data detection using dropout bayesian neural networks. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36. 2022.

36. **Nilsback, M.-E.** and **A. Zisserman**, Automated flower classification over a large number of classes. *In Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*. 2008. URL `https://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html`.

37. **Osada, G.**, **T. Takahashi**, **B. Ahsan**, and **T. Nishide**, Out-of-distribution detection with reconstruction error and typicality-based penalty. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023.

38. **Rahiminasab, Z.**, **M. Yuhas**, and **A. Easwaran**, Out of distribution reasoning by weakly-supervised disentangled logic variational autoencoder. *In 2022 6th International Conference on System Reliability and Safety (ICSRS)*. IEEE, 2022.

39. **Ramakrishna, S.**, **Z. Rahiminasab**, **G. Karsai**, **A. Easwaran**, and **A. Dubey** (2022). Efficient out-of-distribution detection using latent space of β-vae for cyber-physical systems. *ACM Transactions on Cyber-Physical Systems (TCPS)*, **6**(2), 1–34.

40. **Ran, X.**, **M. Xu**, **L. Mei**, **Q. Xu**, and **Q. Liu** (2022). Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation. *Neural Networks*, **145**, 199–208.

41. **Ren, J.**, **P. J. Liu**, **E. Fertig**, **J. Snoek**, **R. Poplin**, **M. Depristo**, **J. Dillon**, and **B. Lakshminarayanan** (2019). Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, **32**.

42. **Rodriguez, E. G.**, On disentanglement and mutual information in semi-supervised variational auto-encoders. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

43. **Segu, M.**, **A. Tonioni**, and **F. Tombari** (2023). Batch normalization embeddings for deep domain generalization. *Pattern Recognition*, **135**, 109115.

44. **Sim, M.**, **J. Lee**, and **H.-J. Choi**, Attention masking for improved near out-of-distribution image detection. *In 2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2023.

45. **Vernekar, S.**, **A. Gaurav**, **V. Abdelzad**, **T. Denouden**, **R. Salay**, and **K. Czarnecki** (2019). Out-of-distribution detection in classifiers via generation. *arXiv preprint arXiv:1910.04241*.

46. **Volpi, R.**, **H. Namkoong**, **O. Sener**, **J. C. Duchi**, **V. Murino**, and **S. Savarese** (2018). Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, **31**.

47. **Wah, C.**, **S. Branson**, **P. Welinder**, **P. Perona**, and **S. Belongie** (2011). The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*. URL `http://www.vision.caltech.edu/visipedia/CUB-200-2011.html`.

48. **Wang, J.**, **C. Lan**, **C. Liu**, **Y. Ouyang**, **T. Qin**, **W. Lu**, **Y. Chen**, **W. Zeng**, and **S. Y. Philip** (2022*a*). Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, **35**(8), 8052–8072.

49. **Wang, X.**, **H. Chen**, **S. Tang**, **Z. Wu**, and **W. Zhu** (2022*b*). Disentangled representation learning.

50. **Wei, H.**, **R. Xie**, **H. Cheng**, **L. Feng**, **B. An**, and **Y. Li**, Mitigating neural network overconfidence with logit normalization. *In International conference on machine learning*. PMLR, 2022.

51. **Xiao, T.**, **J. Hong**, and **J. Ma** (2017). Dna-gan: Learning disentangled representations from multi-attribute images. *arXiv preprint arXiv:1711.05415*.

52. **Xu, J.**, **H. Le**, **M. Huang**, **S. Athar**, and **D. Samaras**, Variational feature disentangling for fine-grained few-shot classification. *In Proceedings of the IEEE/CVF international conference on computer vision*. 2021.

53. **Yang, J.**, **P. Wang**, **D. Zou**, **Z. Zhou**, **K. Ding**, **W. Peng**, **H. Wang**, **G. Chen**, **B. Li**, **Y. Sun**, *et al.* (2022). Openood: Benchmarking generalized out-of-distribution detection. *Advances in Neural Information Processing Systems*, **35**, 32598–32611.

54. **Yang, J.**, **K. Zhou**, **Y. Li**, and **Z. Liu** (2024). Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 1–28.

55. **Zhang, H.**, **Y.-F. Zhang**, **W. Liu**, **A. Weller**, **B. Schölkopf**, and **E. P. Xing**, Towards principled disentanglement for domain generalization. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

56. **Zhang, J.**, **N. Inkawhich**, **R. Linderman**, **Y. Chen**, and **H. Li**, Mixture outlier exposure: Towards out-of-distribution detection in fine-grained environments. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023*a*.

57. **Zhang, J.**, **J. Yang**, **P. Wang**, **H. Wang**, **Y. Lin**, **H. Zhang**, **Y. Sun**, **X. Du**, **K. Zhou**, **W. Zhang**, *et al.* (2023*b*). Openood v1. 5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*.

58. **Zhou, Y.**, Rethinking reconstruction autoencoder-based out-of-distribution detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

59. **Zhu, J.**, **H. Xie**, and **W. Abd-Almageed**, Sw-vae: Weakly supervised learn disentangled representation via latent factor swapping. *In European Conference on Computer Vision*. Springer, 2022.