DISCONET: RETHINKING ADVERSARIAL NETWORKS FOR DISCRIMINATOR-DRIVEN DISTRIBUTION MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Out-of-distribution (OOD) detection holds significant importance across various applications. While semantic and domain-shift OOD problems are well-documented, this work focuses on the nuances of covariate shifts, which entail subtle perturbations or variations in the data distribution. These disturbances have proven to negatively impact machine learning performance. We have found that existing OOD detection methods often struggle to effectively distinguish covariate shifts from in-distribution instances, emphasizing the need for specialized solutions. Therefore, we propose DisCoNet, an Adversarial Variational Autoencoder (VAE) that rethinks the Generative Adversarial Networks paradigm. Instead of prioritizing the generator as the network's core, we focus on the discriminator, using the generator as a supporting training tool. DisCoNet uses the VAE's suboptimal outputs as negative samples to train the discriminator, thereby improving its ability to delineate the boundary between in-distribution samples and covariate shifts. By tightening this in-distribution boundary, DisCoNet achieves state-of-the-art results in public OOD detection benchmarks. The proposed model not only excels in detecting covariate shifts, achieving 98.9% AUROC on ImageNet-1K(-C), but also outperforms all prior methods on public semantic OOD benchmarks. With a model size of $\leq$ 25MB, it is highly effective on Far-OOD (OpenImage-O (99.4%) and iNaturalist (100.0%)) and Near-OOD (SSB-hard (99.9%) and NINCO (99.7%)) detection. The code will be made publicly available.

1 INTRODUCTION

Out-of-distribution (OOD) detection consists of identifying whether a given test sample significantly deviates from the known information of in-distribution (ID) data. It is mainly employed as a preliminary step in image-based systems, aiming to mitigate the risks associated with feeding OOD inputs to a model. Besides safeguarding a system against erroneous predictions, it also facilitates the safe handling of OOD samples, either by rejection or transfer to human intervention. However, the significance of OOD lies not only in bolstering the reliability of image processing systems, but also in its standalone role for anomaly and fault detection. A simple example of this use case can be found in the visual inspection of industrial image data, where it is easy to acquire imagery of normal samples yet virtually impossible to define the expected defects (Roth et al., 2022). In the OOD context, these anomalies can be broadly classified into two types: (1) anomalous objects in images which refer to unexpected or rare items appearing in the frame, and (2) faulty equipment or products which refer to malfunctions or irregularities in the machinery or products under inspection. As a consequence, this task is typically cast as an OOD classification problem.

OOD detection comprises various types of shifts in data. (1) Semantic shifts, such as encountering unseen classes, and (2) domain shifts, like distinguishing between real images and drawings, have easily established boundaries and are well-defined in literature (Hendrycks & Gimpel, 2016; Li et al., 2017). On the other hand, (3) covariate shifts, which involve perturbations in data or subtle changes in its expected variability, are often conflated with domain shifts (Yang et al., 2021). It is essential to differentiate covariate shifts, since they pose unique challenges requiring tailored detection mechanisms.

Figure 1 illustrates the proposed framework for interpreting shifts in a data distribution. In our definition, the ID range covers an expected semantic shift, containing a pre-defined number of different classes, as exemplified by CIFAR-10 (Krizhevsky et al., 2009), along with some degree of variability in terms of domain and covariate shifts. For instance, the introduction of a novel class such as the spiders in ImageNet-200 (Le & Yang, 2015) represents an OOD semantic shift, as CIFAR-10 lacks such examples. An extreme change in domain, such as a hand-drawn representation of a plane from the Sketch dataset (Eitz et al., 2012), is considered OOD, despite the retaining of semantic relevance within the ID set. Additionally, substantial covariate shifts, such as image blurring in the case of a horse image from CIFAR-10(-C) (Hendrycks & Dietterich, 2019), are also classified as OOD, even though there are no explicit alterations in semantic or domain concepts.

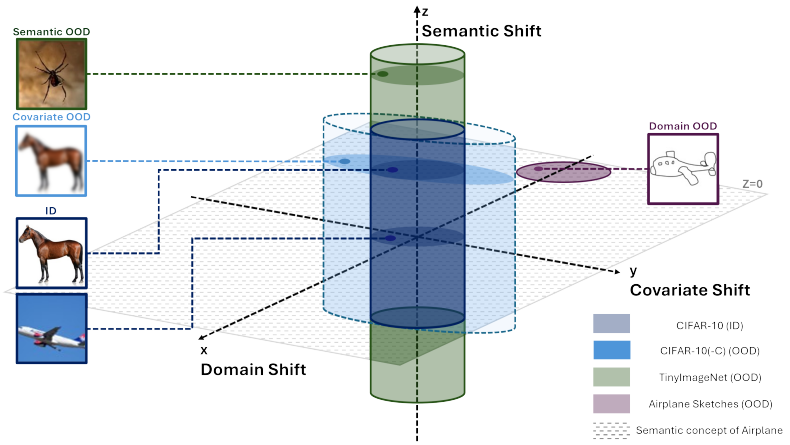

Figure 1: Diagram illustrating data distribution shifts. Variations in the z-axis define semantic shifts, domain shifts represent new contexts like sketches with unchanged semantics and variability, and covariate shifts indicate changes within the same domain and semantic content, such as image perturbations (e.g., blurring).

Various unsupervised OOD detection methods employing generative models like Variational Autoencoders (VAEs) (Pinaya et al., 2021), Generative Adversarial Networks (GANs) (Schlegl et al., 2017), Normalizing Flows (NFs) (Kobyzev et al., 2020) and more recently Denoising Diffusion Probabilistic Models (DDPMs) (Wyatt et al., 2022) have already been explored. The detection of anomalous data is usually performed by assessing whether they deviate from the learned representation manifold, or by comparing the reconstructed and original images in pixel space. DDPMs exhibit superior mode coverage compared to GANs and VAEs, albeit with much slower sampling rates (Xiao et al., 2021). NFs present a good framework for OOD detection, but it is well documented that they often assign a higher likelihood to OOD samples than the ID data (Kirichenko et al., 2020). However, these approaches have focused mainly on semantic and domain-shift detection, rather than covariate shift. Furthermore, the efficiency of these methods is often disregarded.

In this paper, we demonstrate that training a discriminator in an Adversarial VAE framework, using both reconstructed and generated images, results in an excellent OOD detector. Within this covariate shift-focused framework, the model can not only address covariate shift detection but also tackle semantic OOD samples, all while significantly accelerating detection speed compared to prior work. Since the proposed approach is trained end-to-end, the generated sample and image reconstruction quality are refined during training and utilized as OOD samples to improve the discriminator, yielding the quality of these counterfactual examples irrelevant. The main contributions of this work are as follows.

- Redefinition of adversarial training by inverting the roles of the generator and discriminator, using the generator to distill information about in-distribution boundaries, enabling the discriminator to separate in-distribution and out-of-distribution samples effectively.
- DisCoNet, a lightweight unsupervised framework specifically designed for OOD detection.
- State-of-the-art performance in detecting both covariate and semantic OOD shifts, as demonstrated by extensive evaluations that reveal significant improvements over existing methods.

2 Related Work

2.1 Semantic Shift and Covariate Shift OOD

OOD detection literature predominantly focuses on semantic shift and typically falls into two categories: (a) supervised, which requires labels or OOD data, and (b) unsupervised, which relies solely on in-distribution data (Yang et al., 2021). Given the nature of the OOD detection problem, OOD data are typically not available and, as such, unsupervised methods are generally preferred. While the primary aim of this research is covariate shift OOD identification, we provide a brief overview of recent unsupervised advancements in semantic OOD detection that may be useful for covariate OOD detection. Covariate shift occurs when images have consistent semantic and domain content, but are recorded under deviating imaging settings and conditions, or corrupted in a post-processing step. Hence, the degree of variance under these conditions can deteriorate semantic and domain content. This study focuses on covariate shifts within the same domain, as these subtle distribution shifts can cause significant drops in the classification performance of machine learning models, as seen in adversarial attacks (Adhikarla et al., 2023).

2.2 Generative-based Methods for OOD detection

A widely used and initially intuitive approach for OOD detection involves fitting a generative model $p(x;\theta)$ to a data distribution x and evaluating the likelihood of unseen samples under this model, assuming that OOD samples will have lower likelihoods (Bishop, 1994). However, this assumption has been challenged, with various generative models assigning higher likelihoods to certain OOD samples (Hendrycks et al., 2018; Nalisnick et al., 2018). To address this, different approaches have been proposed, including using the Watanabe-Akaike Information Criterion (WAIC) (Choi et al., 2018), specific likelihood ratios (Serrà et al., 2019; Xiao et al., 2020), and hierarchical VAEs (Havtorn et al., 2021). These methods aim to correct for likelihood estimation errors, population-level background statistics, and model feature dominance. Another approach suggests labeling samples as OOD if their likelihoods fall outside the typical range of a model (Chali et al., 2023; Abdi et al., 2024), i.e., a sample may be classified as OOD not only if its likelihood is lower than that of ID data, but also if it is higher (Morningstar et al., 2021).

2.3 Reconstruction-based Methods for OOD detection

Reconstruction-based methods involve training a model R to reconstruct inputs x from the training distribution, such that we obtain $\hat{x} = R(x)$. The rationale is that if R has an information bottleneck it will struggle to accurately reconstruct OOD inputs. However, these methods face practical challenges, including difficulty in tuning the information bottleneck size (Pimentel et al., 2014; Denouden et al., 2018). If it is too small, ID samples may not be faithfully reconstructed; if it is too large, the model can learn the identity function, allowing OOD samples to be reconstructed with low error. Some approaches address these issues by using the Mahalanobis distance in the Autoencoder's feature space as an OOD metric (Denouden et al., 2018), or by introducing a memory module to discourage OOD sample reconstruction (Gong et al., 2019). However, none of these methods fully resolve the bottleneck selection issue. To tackle this limitation, DDPMs have been employed, leveraging noise bottlenecks (Wyatt et al., 2022) and reconstructions from a range of noise values without the need for dataset-specific tuning (Graham et al., 2023) or of corrupted inputs (Liu et al., 2023).

2.4 Feature-based and Logit-based Methods for OOD detection

Several scoring functions have been devised to differentiate between ID and OOD examples, leveraging characteristics of ID samples, but not represented in OOD ones, and vice versa. These functions primarily stem from three sources: (1) probability-based measures, such as maximum softmax probabilities (Hendrycks & Gimpel, 2016), and minimum Kullback-Leibler (KL) divergence between softmax and mean class-conditional distributions (Hendrycks et al., 2019); (2) logit-based functions, including maximum logits (Hendrycks et al., 2019), and the use of the `logsumexp` function computed over logits (Liu et al., 2020); (3) feature-based functions, involving the norm of the residual between a feature and its low-dimensional embeddings (Ndiour et al., 2020), as well as minimum Mahalanobis distance between a feature and class centroids (Lee et al., 2018). Some hybrid methods

combine both logit and feature scores for OOD detection (Wang et al., 2022), while more recent works have introduced masked image modeling pretraining into OOD detection with promising results (Li et al., 2023a; 2024). However, the detection speed of these methods is severely constrained by their transformer-based backbones.

2.5 Adversarial Variational Autoencoders

The VAE (Kingma & Welling, 2013) consists of an encoder that predicts the parameters μ and σ of the variational distribution of the input data, and a decoder that takes a sample from this distribution to reconstruct the input. VAEs are trained to maximize the Evidence Lower Bound (ELBO), which balances reconstruction fidelity with the latent space regularization to ensure that it follows a predefined probability distribution, typically Gaussian. Using latent space as a bottleneck restricts the information that can pass through, leading to uncertainty and blurriness in the reconstructions (Dai & Wipf, 2019). Additionally, the pixel-wise reconstruction error and the high dimensionality of natural image manifolds pose challenges for VAEs in generating high-quality and realistic samples. While natural images are assumed to lie on low-dimensional manifolds due to local scale redundancy (Kretzmer, 1952), textures exist in higher-dimensional manifolds, making them difficult to capture.

GANs (Goodfellow et al., 2014) consist of two neural networks with adversarial objectives: the generator learns to map a random vector to the data space; the discriminator acts as a classifier trained to differentiate real samples from generated ones. Despite their success in generation tasks, GANs suffer from two primary limitations compared to VAEs. The first is mode collapse, which occurs when the generator produces only a few different types repeatedly, making it easily recognizable by the discriminator. Consequently, the discriminator's feedback lacks useful information (Thanh-Tung & Tran, 2020). Additionally, GANs lack an encoder network, which limits their ability to perform reconstruction and latent space manipulation.

The VAE and GAN have been combined by incorporating a discriminator to enhance the realism of VAE reconstructions (Larsen et al., 2016). Alternatively, the BiGAN (Donahue et al., 2016) architecture features an encoder, generator, and discriminator, aiming for good unsupervised feature representations but tends to produce less accurate reconstructions. Other approaches have adapted this VAE/GAN combination to fully utilize the strengths of each architecture to improve the realism of the images produced by the model (Plumerault et al., 2021). The objective of DisCoNet is to retain the adversarial benefits and the mode coverage of the hybrid strategy, without the final goal of image generation, thereby reducing computational requirements.

3 DisCoNet

3.1 Overview

A prevalent generative-based approach to OOD detection involves leveraging a trained model to assess the likelihood of new, unseen samples. Similarly, an adversarially trained discriminator can provide a boundary for the ID set, by assessing the probability of a sample being real (ID) or synthetic (OOD). By adjusting where the discriminator learns to draw this boundary, we can create an OOD detector.

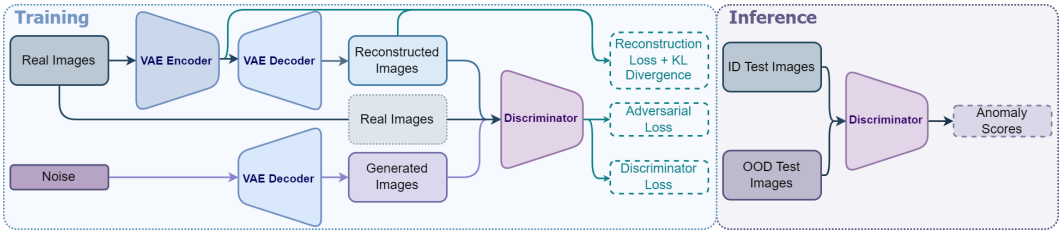

Figure 2: Overview of DisCoNet. During inference, only the Discriminator is used.

It is on this premise that we propose a **Dis**crimative **Co**variate Shift **Net**work, DisCoNet. DisCoNet is an Adversarial VAE-inspired architecture, as shown in Figure 2, in which both the VAE and the discriminator are trained adversarially. DisCoNet's approach combines generative and reconstruction-based strategies to distill information about the ID set and OOD boundaries to the discriminator

during training in an unsupervised manner. Unlike traditional adversarial methods, DisCoNet's focus is on leveraging the generator's output as a tool to refine the discriminator.

The VAE is trained to reduce the standard ELBO loss, while also producing samples (generated images using the VAE decoder) that can fool the discriminator. The discriminator is trained to not only distinguish between generated and real images, as in the standard GAN setup, but also reconstructed images. Reconstructions from VAEs typically lack detail, i.e., they have a sub-optimal high-frequency representation (Lin et al., 2023), which can be found in certain types of covariate shifts, such as blurriness. On the other hand, images generated from GANs often exhibit severe high-frequency differences, leading the discriminator to focus excessively on these components (Li et al., 2023b). This focus can hinder the generator's ability to capture low-frequency components. By training the discriminator on reconstructions and generations, and encouraging both to appear more realistic, the discriminator's boundaries of the ID frequency spectrum become tighter, strengthening its ability to detect OOD samples, as illustrated in Figure 3.

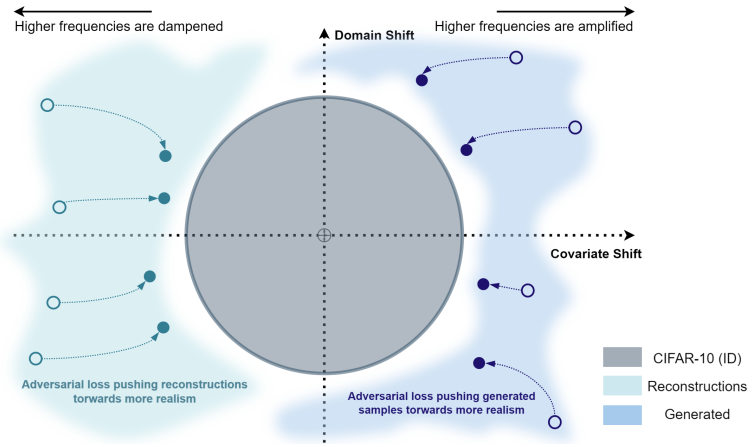

Figure 3: Covariate shifts can be simulated by reconstructed and generated images. Encouraging more realism helps to tighten the border between the ID and the OOD sets.

3.2 Training

The VAE in DisCoNet's framework remains unchanged compared to the traditional VAE, with parameters θ and composed of an encoder $\mathcal{E}_{\theta_\text{E}}$ and a decoder $\mathcal{G}_{\theta_\text{G}}$ responsible for generating an image output. The VAE is a parameterized model given by $q_{\theta_\text{E}}(z|x^{(i)}) = \mathcal{N}(z; \mu^{(i)}, \sigma^{2(i)}\mathbf{I})$, where $\mu^{(i)}$ and $\sigma^{2(i)}$ are outputs of $\mathcal{E}_{\theta_\text{E}}$. The prior distribution of the latent codes is $p(z) = \mathcal{N}(z; 0, \mathbf{I})$. The VAE loss function combines a reconstruction term and a latent space regularization term, as demonstrated in the original paper by Kingma & Welling (2013) and in adversarial implementations (Plumerault et al., 2021). The reconstruction term optimizes the encoding-decoding process, while the regularization term aligns the encoder distributions with a standard normal distribution. The latter is represented by the KL-divergence between the predicted distribution and a standard Gaussian. Both terms are represented in Figure 2 and can be written as

$$\mathcal{L}_{\text{VanillaVAE}} = \|x^{(i)} - \mathcal{G}_{\theta_\text{G}}(z)\|^2 - \frac{1}{2}\sum_{j=1}^{\dim(z)}\left(1 + \log\left(\sigma_j^{2(i)}\right) - \mu_j^{2(i)} - \sigma_j^{2(i)}\right). \tag{1}$$

An additional model, the discriminator $\mathcal{D}$, parameterized by ϕ, is added to the traditional VAE architecture. It has two main goals, as shown in Figure 2. First, it must discern between real images and images either reconstructed from z_real or generated from random noise z_fake. This can be achieved by minimizing the cross-entropy function

$$\begin{aligned}\mathcal{L}_\text{D} = &\mathbb{E}_{x\sim p(x)}\left[\log\left(1 - \mathcal{D}_\phi(x)\right)\right] \\ &+ \mathbb{E}_{x\sim p_{\theta_\text{G}}(x|z_\text{real})}\left[\log\left(\mathcal{D}_\phi(x)\right)\right] + \mathbb{E}_{x\sim p_{\theta_\text{G}}(x|z_\text{fake})}\left[\log\left(\mathcal{D}_\phi(x)\right)\right].\end{aligned} \tag{2}$$

This suggests that in addition to the discriminator's initial goal of improving generated images (sampled from random noise), it also pushes the reconstructions toward more realism. Therefore,

an adversarial loss term, which encourages the VAE to generate or reconstruct images that fool the discriminator, is added to the loss function, so that

$$\mathcal{L}_{\text{Adv}} = \mathbb{E}_{x\sim p_{\theta_G}(x|z_{\text{real}})}\left[1 - \log\left(\mathcal{D}_\phi(x)\right)\right] + \mathbb{E}_{x\sim p_{\theta_G}(x|z_{\text{fake}})}\left[1 - \log\left(\mathcal{D}_\phi(x)\right)\right]. \tag{3}$$

The final DisCoNet loss function is thus a weighted combination of both the Vanilla VAE loss and the adversarial loss, which results in

$$\begin{aligned}\mathcal{L}_{\text{Total}} = \|x^{(i)} - \mathcal{G}_{\theta_G}(z)\|^2 - \frac{\omega_{\text{KL}}}{2}\sum_{j=1}^{\dim(z)}\left(1 + \log\left(\sigma_j^{2(i)}\right) - \mu_j^{2(i)} - \sigma_j^{2(i)}\right) \\ + \omega_{\text{Rec}}\mathbb{E}_{x\sim p_{\theta_G}(x|z_{\text{real}})}\left[1 - \log\left(\mathcal{D}_\phi(x)\right)\right] + \omega_{\text{Gen}}\mathbb{E}_{x\sim p_{\theta_G}(x|z_{\text{fake}})}\left[1 - \log\left(\mathcal{D}_\phi(x)\right)\right].\end{aligned} \tag{4}$$

3.3 Inference

During test time, only the discriminator is utilized through a single forward pass across the network to determine the probability of a sample belonging to the ID set. However, we aim to express the results as anomaly scores $\mathcal{A}$, ideally attributing an anomaly score of 0 to an ID sample, and a score of 1 to an OOD sample. This score can be defined as

$$\mathcal{A}(x) = -\mathcal{D}_\phi(x) + 1. \tag{5}$$

4 Benchmark Methodology

4.1 Datasets

In OOD detection benchmarks, the conventional approach involves designating an entire dataset as ID and then compiling multiple datasets that lack any semantic overlap with the ID categories to act as OOD sets.

To ensure consistency in the benchmarking process, we adhere to the methodology proposed by OpenOOD (Yang et al., 2022). Our evaluation encompasses three tasks: (1) *Near-OOD*, which exhibits slight semantic variation compared to ID datasets; (2) *Far-OOD*, which encompasses both semantic and domain shifts; and (3) *Covariate Shift OOD*, involving corruptions within the ID set. Three datasets are defined as ID: CIFAR-10 (Krizhevsky et al., 2009), ImageNet-200 (Le & Yang, 2015), and ImageNet-1K (Russakovsky et al., 2015). Further details on dataset selection and availability are summarized in Appendix A.1.

4.2 Evaluation Metrics

The evaluation metrics employed in OpenOOD by Yang et al. (2022) are adopted for this work. These two main evaluation metrics are: (1) *AUROC*, which measures the area under the Receiver Operating Characteristic (ROC) curve, and displays the relationship between True Positive Rate (TPR) and False Positive Rate (FPR); and (2) *FPR@95*, which measures the FPR when the TPR is equal to 95%, with lower scores indicating better performance. The full results are provided in the form "AUROC/FPR@95%".

4.3 Selected Models

A set of SOTA models representing the various approaches to OOD detection are trained and employed as baselines for extensive validation of the proposed approach. *GLOW* (Kingma & Dhariwal, 2018) represents generative-based methods, allowing assessment through both Log-Likelihood and the recently proposed typicality (Chali et al., 2023) metric. *DDPM-OOD* (Graham et al., 2023) is the selected reconstruction-based technique. For feature-based and logit-based methodologies, we opt for *MOODv2* (Li et al., 2024), a state-of-the-art model for semantic OOD detection. To explore adversarial techniques, a *Deep Convolutional GAN (DC-GAN)* (Radford et al., 2015) provides a baseline, alongside a *Prescribed GAN* (Dieng et al., 2019) to investigate the impact of mode collapse mitigation strategies. For the ImageNet-1K benchmark, we compare against the available public models instead of retraining models that have shown limited effectiveness. Specifically, we evaluate

our method against *MOODv2*, *NNGuide* (Park et al., 2023), and *SCALE* (Xu et al., 2023), which report SOTA performance on Near-OOD and Far-OOD detection for these datasets. The implementation details are provided in Appendix A.2 and the source code will be publicly released.

5 Experiments & Results

This section describes the conducted experiments and presents the key results. It covers Covariate Shift OOD detection, as well as Near-OOD and Far-OOD detection performance for the selected models. Additional detailed experimental results can be found in the supplemental materials.

5.1 Covariate Shift OOD

As shown in Table 1, DisCoNet consistently outperforms all other models in detecting OOD covariate shifts across CIFAR-10, ImageNet-200, and ImageNet-1K, achieving the highest average detection scores across all corruption intensities. DDPM-OOD shows improved detection on ImageNet-200 compared to CIFAR-10, while MOODv2 demonstrates a decrease in OOD detection. NNGuide demonstrates robust performance for higher intensities. SCALE's results are not reported due to difficulties in implementing Covariate Shift within their framework. Detailed performance analyses for each model, including per corruption type and intensity, can be found in Appendices A.4, A.5, and A.6.

Table 1: Covariate shift OOD benchmark results for models trained on CIFAR-10 and ImageNet-200.

ID	Model	Corruption Intensity					Average
		1	2	3	4	5	
CIFAR-10	GLOW (Kingma & Dhariwal, 2018)	60.7/71.9	57.5/71.5	58.4/69.5	58.7/68.4	58.7/66.3	58.9/69.5
	GLOW (Chali et al., 2023) (Typ.)	41.9/90.8	42.9/85.5	41.2/86.8	40.7/84.8	41.2/81.2	41.6/85.8
	DDPM-OOD (Graham et al., 2023)	59.1/88.5	64.3/81.7	68.7/73.5	71.3/70.6	75.3/63.0	67.8/75.5
	MOODv2 (Li et al., 2024)	72.0/82.1	74.9/78.8	76.9/76.1	78.7/73.8	82.1/69.2	76.9/76.0
	DC-GAN (Radford et al., 2015)	52.6/92.8	53.9/91.5	54.9/90.9	55.7/90.2	56.6/89.3	54.7/90.9
	PresGAN (Dieng et al., 2019)	63.1/86.9	70.9/78.2	73.2/70.8	75.8/65.4	79.5/60.0	72.5/72.3
	DisCoNet (Prop.)	**90.0/30.3**	**97.4/9.2**	**96.0/10.1**	**98.2/5.4**	**99.3/2.0**	**96.2/11.4**
ImageNet-200	GLOW (Kingma & Dhariwal, 2018)	35.2/91.0	38.4/81.9	37.0/79.0	35.8/78.4	34.7/78.4	36.2/81.7
	GLOW (Chali et al., 2023) (Typ.)	50.6/86.0	48.8/82.8	49.9/78.9	51.7/75.0	53.8/71.4	51.0/78.8
	DDPM-OOD (Graham et al., 2023)	67.6/75.5	71.7/69.6	76.8/60.3	79.5/52.5	81.9/48.4	75.5/61.3
	MOODv2 (Li et al., 2024)	59.8/89.1	62.8/87.3	67.1/84.2	72.2/80.5	77.1/76.3	67.8/83.5
	DC-GAN (Radford et al., 2015)	57.6/87.7	58.7/86.4	59.9/84.9	61.2/83.3	62.1/81.9	59.9/84.8
	PresGAN (Dieng et al., 2019)	61.2/88.5	64.3/86.1	68.1/80.4	70.6/74.1	70.9/70.1	67.0/79.9
	DisCoNet (Prop.)	**99.7/1.9**	**99.7/1.7**	**99.7/2.0**	**99.8/1.3**	**99.8/0.8**	**99.7/1.5**
IN-1K	MOODv2 (Li et al., 2024)	60.3/88.9	65.4/82.8	69.7/76.2	75.3/67.4	81.8/54.3	70.5/73.9
	NNGuide (Park et al., 2023)	64.3/82.0	72.3/71.9	78.9/61.2	86.0/46.2	91.4/32.0	78.6/58.7
	DisCoNet (Prop.)	**97.3/9.2**	**98.6/5.1**	**99.2/2.8**	**99.7/1.2**	**99.8/0.8**	**98.9/3.8**

5.2 Near-OOD and Far-OOD

Table 2: Near-OOD and Far-OOD detection benchmark results for models trained on CIFAR-10.

Model	Near-OOD		Far-OOD			
	CIFAR-100	TIN	MNIST	SVHN	DTD	Places365
GLOW (Kingma & Dhariwal, 2018)	51.7/96.2	48.5/94.4	0.0/100.0	7.2/99.3	62.8/96.8	66.3/91.9
GLOW (Chali et al., 2023) (Typ.)	47.4/97.0	65.6/81.5	**100.0/0.0**	91.3/19.3	29.5/99.9	25.3/98.6
DDPM-OOD (Graham et al., 2023)	56.5/92.6	63.0/87.5	31.6/98.7	95.8/28.1	88.1/59.5	62.8/85.9
MOODv2 (Li et al., 2024)	**89.2/45.7**	**96.1/18.6**	99.1/0.7	97.0/18.2	**100.0/0.0**	**99.9/0.2**
DC-GAN (Radford et al., 2015)	50.1/97.0	55.5/94.4	42.2/95.4	62.9/87.4	61.4/93.8	58.0/91.6
PresGAN (Dieng et al., 2019)	51.6/94.3	55.4/93.0	24.8/98.7	87.0/51.6	34.0/99.6	65.9/86.5
DisCoNet (Prop.)	75.0/75.4	91.4/37.6	**100.0/0.0**	**100.0/0.0**	66.2/97.3	92.6/34.0

Table 2 presents the Near-OOD and Far-OOD detection scores, obtained by the selected models when trained with CIFAR-10 as the ID set. MOODv2 shows a clear advantage in Near-OOD tasks and delivers consistent results in Far-OOD scenarios, almost matching DisCoNet in the few datasets where it is not the best performer. Among the adversarial methods, DisCoNet achieves the best results across all datasets, although its performance on CIFAR-100 and DTD is less impressive.

Table 3 depicts the Near-OOD and Far-OOD scores achieved by the selected models when trained on ImageNet-200 as the ID set. In this case, DisCoNet emerges as the clearly best-performing approach overall, illustrating image scaling benefits. MOODv2 comes close behind. Moreover, the GLOW model trained on Log-Likelihood also performs significantly better when trained at higher resolutions.

Table 3: Near-OOD and Far-OOD detection benchmark results for models trained on ImageNet-200.

Model	Near-OOD		Far-OOD		
	SSB-hard	NINCO	iNaturalist	DTD	OpenImage-O
GLOW (Kingma & Dhariwal, 2018)	84.5/88.7	81.7/97.0	99.2/0.5	87.7/84.0	88.5/88.7
GLOW (Chali et al., 2023) (Typ.)	65.0/94.0	69.5/94.3	49.6/98.7	66.0/99.9	56.4/99.8
DDPM-OOD (Graham et al., 2023)	61.4/89.9	64.7/90.4	54.3/88.0	61.0/99.3	50.7/96.6
MOODv2 (Li et al., 2024)	96.6/12.7	99.4/2.0	**100.0**/0.1	99.0/4.3	99.8/0.8
DC-GAN (Radford et al., 2015)	55.6/94.4	56.8/94.3	45.3/95.4	54.4/98.0	50.7/98.2
PresGAN (Dieng et al., 2019)	57.3/93.2	48.9/96.5	92.6/41.7	23.1/99.7	61.7/92.7
DisCoNet (Prop.)	**100.0/0.0**	**100.0/0.0**	**100.0/0.0**	**99.5/0.9**	**100.0/0.0**

Table 4 illustrates that the image scaling benefits observed in ImageNet-200 persist in ImageNet-1K, with DisCoNet outperforming SOTA methods in 4 out of 5 benchmarks. However, as noted in the CIFAR-10 experiments, DisCoNet exhibits some inconsistencies when evaluated on the DTD dataset.

Table 4: Near-OOD and Far-OOD detection benchmark results for models trained on ImageNet-1K. Results are obtained from the respective research papers where available or recomputed (*) and reported following the OpenOODv1.5 benchmark.

Model	Near-OOD		Far-OOD		
	SSB-hard	NINCO	iNaturalist	DTD	OpenImage-O
MOODv2 (Li et al., 2024)	85.0/58.1*	92.7/38.2*	99.6/1.8	94.3/24.7	97.4/13.6
SCALE (Xu et al., 2023)	77.4/67.7	85.4/51.8	98.0/9.5	97.6/11.9	94.0/28.2
NNGuide (Park et al., 2023)	84.7/54.7*	93.7/28.9*	99.9/1.8	**99.4/17.0**	99.1/10.8
DisCoNet (Prop.)	**99.9/0.0**	**99.7/0.1**	**100.0/0.0**	87.6/84.0	**99.4/0.3**

5.3 DisCoNet Ablation Study

To evaluate the impact of using both reconstructions and generated images for training DisCoNet's discriminator, we have performed an ablation experiment in which we train models using (1) only reconstructions, (2) only with generated images, and (3) both reconstructed and generated images. The corruptions applied to images in CIFAR-10(-C) and ImageNet-200(-C) shift their frequency spectrum. This was used to propose splitting the corruptions into two major groups: *Lower Frequency* refers to corruptions that reduce high-frequency components (e.g., Gaussian Blur), while *Higher Frequency* refers to those that increase high-frequency components (e.g., Impulse Noise). Appendix A.7 provides experimental evidence of their impact on the frequency spectrum, while Table 30 displays the proposed split.

As anticipated, Table 5 indicates that models trained solely on reconstructions excel at detecting low-frequency corruptions, but struggle with high-frequency ones. In contrast, models trained on generated images perform better at identifying high-frequency corruptions. Interestingly, these models demonstrate a good ability to detect low-frequency corruptions, indicating that this approach covers a broader spectrum of perturbations. Detailed results for every corruption are provided in Appendix A.8.

Table 5: Ablation study showing the impact of using reconstructed or generated images during DisCoNet's training.

Corruption	CIFAR-10(-C)			ImageNet-200(-C)		
	Recon.	Generated	Both	Recon.	Generated	Both
Lower Frequency	96.5/12.1	75.7/66.4	97.5/8.7	100.0/0.0	97.1/11.5	100.0/0.0
Higher Frequency	24.0/94.1	81.1/52.2	95.2/13.4	92.8/11.0	98.1/8.1	99.5/2.9
Average	54.6/59.6	78.8/58.2	96.2/11.4	96.2/5.8	97.6/9.7	99.7/1.5

5.4 Adversarial Attacks

To test the model's robustness to adversarial attacks, we follow Azizmalayeri et al. (2022) and employ both black box (Gaussian Noise) and white box (Projected Gradient Descent, PGD) approaches, adapting the `torchattacks` library for single-channel binary OOD prediction. These attacks are applied to the model trained on ImageNet-200, and its performance is assessed on the ImageNet-200 ID test set versus Near-OOD test sets.

Table 6: Adversarial attack on the ImageNet-200-trained DisCoNet. Attacks on the ID set vs. the Near OOD test sets (Average of SSB-hard and NINCO) are compared.

ID Attack vs.	OOD Clean	OOD PGD 5	OOD PGD 10	OOD PGD 20	GN σ 0.1	GN σ $\frac{8}{255}$
Clean	100.0/0.0	100.0/0.0	85.5/54.5	19.6/99.7	100.0/0.0	100.0/0.1
PGD 1	100.0/0.0	100.0/0.0	75.1/73.6	11.1/99.9	100.0/0.0	100.0/0.0
PGD 5	100.0/0.0	100.0/0.0	75.1/73.8	11.0/99.9	100.0/0.0	100.0/0.0
PGD 10	100.0/0.0	100.0/0.0	75.0/73.8	11.0/99.9	100.0/0.0	100.0/0.0
GN σ 0.1	99.9/0.2	99.9/0.2	0.1/100.0	0.0/100.0	99.9/0.2	99.9/0.2
GN σ $\frac{8}{255}$	100.0/0.0	100.0/0.0	57.6/90.2	4.3/99.9	100.0/0.0	100.0/0.0

As shown in Table 6, the model attains near-perfect AUROC across all black box attacks, which are treated as covariate shifts. Even the weakest Gaussian Noise perturbation (STD 8/255) introduces less variation than Severity 1 in ImageNet-200(-C). For white box attacks, we employ PGD with default settings (Mądry et al., 2017), increasing the number of steps to intensify the attack. The model remains robust up to 5 steps, with performance degradation observed only at higher iterations. The largest performance drop occurs when ID data is perturbed by Gaussian Noise (treated as OOD in prior experiments) and OOD data undergoes over 10 PGD steps. Although this scenario is unrealistic, the model exhibits strong overall robustness to standard adversarial threats.

6 Discussion

Covariate Shift OOD. The extensive validation across multiple datasets and models demonstrates that DisCoNet outperforms the other models in detecting Covariate Shifts by a large margin. The results indicate that relying solely on a GAN, even with enhancements from PresGAN aiming at mitigating mode collapse, is insufficient to achieve high performance. In Section 3, we have hypothesized, supported by literature findings (Lin et al., 2023; Li et al., 2023b), that training a discriminator with VAE reconstructions would enhance its sensitivity to corruptions that dampen the high-frequency spectrum. Alternatively, training a discriminator with generated images improves its ability to detect high-frequency amplification. The ablation study results in Table 5 validate the hypothesis from Section 3. Models trained exclusively on reconstructed images, which are typically blurrier, present significantly higher classification scores in identifying low-frequency corrupted images as OOD compared to high-frequency ones. This inability to detect high-frequency corruptions is particularly evident in CIFAR-10(-C). This behavior can be explained easily: reconstructed images lack significant high-frequency components, leading the discriminator to simply learn to classify images with low high-frequency content as fake and images with high-frequency content as real. This is further supported by the AUROC scores achieved on the noise-related perturbations in CIFAR-10(-C), where the model erroneously considers these samples closer to being ID than the actual ID samples. Conversely, models trained on generated images perform better in detecting high-frequency

corruptions, as expected. Surprisingly, their ability to detect low-frequency corruptions is remarkably high, indicating that using this method covers a broader spectrum of perturbations.

Near-OOD and Far-OOD. DisCoNet performs well in both Near-OOD and Far-OOD detection scenarios, outperforming other approaches when using ImageNet-200 as the ID dataset and continuing to consistently rank first in the ImageNet-1K analysis. Additionally, in the CIFAR-10 benchmark, DisCoNet ranks second only to MOODv2 in various Far-OOD tests; CIFAR-10 poses a greater challenge for DisCoNet because its pixelated resolution inherently introduces corruptions, making it harder for the model to assign low scores to ID samples.

Deployment Scenarios. There are generally two main deployment scenarios for OOD detection algorithms: (1) The OOD detection algorithm is the primary focus, deployed as a standalone application. (2) The OOD detection algorithm operates alongside a main image processing algorithm, ensuring its safe and effective use. An OOD algorithm must be practical and effective in real-world scenarios, delivering strong detection performance while being highly deployable. Deployability should be assessed in the following four critical areas.

1. Accessibility: Evaluated by the compute requirements necessary for the algorithm.
2. Development Cycle: Measured by the time required for model training and deployment.
3. Inference Speed: The time it takes for the algorithm to make predictions during deployment.
4. Accuracy: The ability of the algorithm to provide highly accurate OOD detection.

An ideal OOD detection algorithm excels in all the above dimensions, ensuring it can be effectively utilized in various practical applications. As detailed in our results and Appendix A.3, DisCoNet excels in all criteria. The model achieves SOTA OOD detection results, while utilizing substantially smaller and faster models. Table 12 indicates that DisCoNet is up to 3 orders of magnitude faster than MOODv2 and up to 4 orders of magnitude quicker than GLOW and the DDPM-OOD. Training is also much more efficient than GLOW and DDPM-OOD, as demonstrated in Table 11, making it a strong candidate for applications that require fast development cycles.

7 Limitations & Future work

One potential limitation of DisCoNet is that, while it has demonstrated promising results on 128×128 resolution images, further evaluation on higher resolutions is required to fully assess its scalability. Although performance improvements have been observed with increasing resolution, additional experiments with larger image sizes are necessary to draw definitive conclusions. Future research should prioritize applying DisCoNet to high-resolution domains. Additionally, with regard to covariate shift detection, only three models have been evaluated on ImageNet-1K to date, highlighting the need for broader benchmarking. Despite DisCoNet's strong performance, future efforts will focus on ensuring a more extensive comparison. Future work will focus on gaining a deeper understanding of DisCoNet's internal mechanisms, particularly how it processes changes in the frequency spectrum from a signal processing perspective.

8 Conclusion

This paper introduces DisCoNet, a novel approach for OOD detection, with a focus on covariate shift detection. Contrary to standard VAE or GAN training objectives, DisCoNet utilizes a combination of both reconstructed and generated images to address a broad range of frequency-spectrum perturbations for improved performance. To the best of our knowledge, the model demonstrates state-of-the-art OOD detection performance, achieving an AUROC of 96.2% on CIFAR-10(-C), 99.7% on ImageNet-200(-C) and 98.9% on ImageNet-1k(-C). Additionally, DisCoNet shows excellent performance in semantic OOD detection tasks, surpassing or matching current SOTA methods such as MOODv2, NNGuide and SCALE on ImageNet-200 and ImageNet-1K. DisCoNet's lightweight and fast architecture, along with its efficient training cycles, make it a practical choice for real-world and real-time applications requiring low computational resources.

REFERENCES

Lemar Abdi, Amaan Valiuddin, Christiaan Viviers, Fons van der Sommen, et al. Typicality excels likelihood for unsupervised out-of-distribution detection in medical imaging. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging-6th International Workshop*, 2024.

Eashan Adhikarla, Kai Zhang, Jun Yu, Lichao Sun, John Nicholson, and Brian D Davison. Robust computer vision in an ever-changing world: A survey of techniques for tackling distribution shifts. *arXiv preprint arXiv:2312.01540*, 2023.

Mohammad Azizmalayeri, Arshia Soltani Moakhar, Arman Zarei, Reihaneh Zohrabi, Mohammad Manzuri, and Mohammad Hossein Rohban. Your out-of-distribution detection method is not robust! *Advances in Neural Information Processing Systems*, 35:4887–4901, 2022.

Christopher M Bishop. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994.

Samy Chali, Inna Kucher, Marc Duranton, and Jacques-Olivier Klein. Improving normalizing flows with the approximate mass for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 750–758, 2023.

Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

Taylor Denouden, Rick Salay, Krzysztof Czarnecki, Vahdat Abdelzad, Buu Phan, and Sachin Vernekar. Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance. *arXiv preprint arXiv:1812.02765*, 2018.

Adji B Dieng, Francisco JR Ruiz, David M Blei, and Michalis K Titsias. Prescribed generative adversarial networks. *arXiv preprint arXiv:1910.04302*, 2019.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):44:1–44:10, 2012.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1705–1714, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Mark S Graham, Walter HL Pinaya, Petru-Daniel Tudosiu, Parashkev Nachev, Sebastien Ourselin, and Jorge Cardoso. Denoising diffusion models for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2947–2956, 2023.

Jakob D Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical vaes know what they don't know. In *International Conference on Machine Learning*, pp. 4117–4128. PMLR, 2021.

Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.

Rui Huang and Yixuan Li. Mos: Towards scaling out-of-distribution detection for large semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8710–8719, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43 (11):3964–3979, 2020.

Ernest R Kretzmer. Statistics of television signals. *The bell system technical journal*, 31(4):751–763, 1952.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pp. 1558–1566. PMLR, 2016.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Jingyao Li, Pengguang Chen, Zexin He, Shaozuo Yu, Shu Liu, and Jiaya Jia. Rethinking out-of-distribution (ood) detection: Masked image modeling is all you need. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11578–11589, 2023a.

Jingyao Li, Pengguang Chen, Shaozuo Yu, Shu Liu, and Jiaya Jia. Moodv2: Masked image modeling for out-of-distribution detection. *arXiv preprint arXiv:2401.02611*, 2024.

Ziqiang Li, Pengfei Xia, Xue Rui, and Bin Li. Exploring the effect of high-frequency components in gans training. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(5):1–22, 2023b.

Xinmiao Lin, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Yu Kong. Catch missing details: Image reconstruction with frequency augmented variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1736–1745, 2023.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Zhenzhen Liu, Jin Peng Zhou, Yufan Wang, and Kilian Q Weinberger. Unsupervised out-of-distribution detection with diffusion inpainting. In *International Conference on Machine Learning*, pp. 22528–22538. PMLR, 2023.

Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017.

Warren Morningstar, Cusuh Ham, Andrew Gallagher, Balaji Lakshminarayanan, Alex Alemi, and Joshua Dillon. Density of states estimation for out of distribution detection. In *International Conference on Artificial Intelligence and Statistics*, pp. 3232–3240. PMLR, 2021.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

Ibrahima Ndiour, Nilesh Ahuja, and Omesh Tickoo. Out-of-distribution detection with subspace techniques and probabilistic modeling of features. *arXiv preprint arXiv:2012.04250*, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, Spain, 2011.

Jaewoo Park, Yoon Gyo Jung, and Andrew Beng Jin Teoh. Nearest neighbor guidance for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1686–1695, 2023.

Z Peng, L Dong, H Bao, Q Ye, and F Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. arxiv 2022. *arXiv preprint arXiv:2208.06366*.

Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

Walter Hugo Lopez Pinaya, Petru-Daniel Tudosiu, Robert Gray, Geraint Rees, Parashkev Nachev, Sébastien Ourselin, and M Jorge Cardoso. Unsupervised brain anomaly detection and segmentation with transformers. *arXiv preprint arXiv:2102.11650*, 2021.

Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Avae: adversarial variational auto encoder. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8687–8694. IEEE, 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pp. 146–157. Springer, 2017.

Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480*, 2019.

Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pp. 1–10. IEEE, 2020.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need? 2021.

Christiaan Viviers, Amaan Valiuddin, Francisco Caetano, Lemar Abdi, Lena Filatova, Peter de With, and Fons van der Sommen. Can your generative model detect out-of-distribution covariate shift? *arXiv preprint arXiv:2409.03043*, 2024.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4921–4930, 2022.

Julian Wyatt, Adam Leach, Sebastian M. Schmon, and Chris G. Willcocks. Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 650–656, June 2022.

Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *Advances in neural information processing systems*, 33:20685–20696, 2020.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Kai Xu, Rongyu Chen, Gianni Franchi, and Angela Yao. Scaling for training time and post-hoc out-of-distribution detection enhancement. *arXiv preprint arXiv:2310.00227*, 2023.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, et al. Openood: Benchmarking generalized out-of-distribution detection. *Advances in Neural Information Processing Systems*, 35:32598–32611, 2022.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A APPENDIX/SUPPLEMENTAL MATERIAL

The supplementary material is organized as follows: Appendix A.1 covers the datasets used in this work. Appendix A.2 describes the implementation details of the employed models while Appendix A.3 covers the compute resources required for training and evaluating the models. Appendix A.4 provides detailed results on the CIFAR-10 Covariate Shift OOD detection benchmark, while results for ImageNet-200 and ImageNet-1K are covered in Appendix A.5 and Appendix A.6, respectively. Appendix A.7 explains the impact of each corruption in the frequency spectrum. Finally, a series of ablation experiments are provided in Appendix A.8.

A.1 DATA AVAILABILITY

Three datasets are defined as ID: CIFAR-10 (Krizhevsky et al., 2009), ImageNet-200 (Le & Yang, 2015), and ImageNet-1K (Russakovsky et al., 2015).

CIFAR-10 is a 10-class general object classification dataset with 50k training and 10k test images. Near-OOD is assessed using CIFAR-100 (Krizhevsky et al., 2009) and ImageNet-200. The Far-OOD benchmark includes MNIST, SVHN (Netzer et al., 2011), DTD, and Places365. Covariate Shift is evaluated with CIFAR-10(-C) (Hendrycks & Dietterich, 2019).

ImageNet-200 is a subset of ImageNet with 200 classes. For Near-OOD, SSB-hard (Vaze et al., 2021) and NINCO (He et al., 2021) are used. Far-OOD includes iNaturalist (Huang & Li, 2021), DTD, and OpenImage-O (Wang et al., 2022). Covariate Shift is evaluated using ImageNet-200(-C) (Hendrycks & Dietterich, 2019).

ImageNet-1K contains 1000 classes. The Near-OOD and Far-OOD datasets correspond to those used for ImageNet-200. Covariate Shift is evaluated using ImageNet-1K(-C) (Hendrycks & Dietterich, 2019).

The OOD benchmark used to evaluate and compare the selected models closely follows the one proposed in OpenOOD by Yang et al. (2022), which contains all the datasets mentioned in Table 7, except for CIFAR-10(-C).

Table 7: Datasets used for the OOD benchmark. Legend: *dataset not present in OpenOOD.

ID	Near-OOD	Far-OOD	Covar. Shift OOD	Resolution
CIFAR-10	CIFAR-100, TIN	MNIST, SVHN, DTD, Places365	CIFAR-10(-C)*	32×32 px
ImageNet-200	SSB-hard, NINCO	iNaturalist, DTD, OpenImage-O	ImageNet-200(-C)	64×64 px
ImageNet-1K	SSB-hard, NINCO	iNaturalist, DTD, OpenImage-O	ImageNet-1K(-C)	128×128 px

CIFAR-10(-C), ImageNet-200(-C), and ImageNet-1K(-C) were downloaded from their source [1]. Additionally, we used the original and publicly available splits for ImageNet-200 [2] and ImageNet-1K [3]. The remaining datasets and files containing training and evaluation splits were downloaded from OpenOOD's publicly available repository [4]. For convenience, DisCoNet's repository includes a script that automatically downloads these datasets and contains the split files.

A.2 IMPLEMENTATION DETAILS

This appendix provides the required implementation details of the employed models for reproducibility.

A.2.1 GLOW

Utilizing Normalizing Flows for OOD detection involves modeling the ID data distributions through invertible transformations, maximizing the log-likelihood training objective. In this study, we utilize

[1] https://github.com/hendrycks/robustness
[2] https://www.kaggle.com/datasets/nikhilshingadiya/TinyImageNet200
[3] https://www.kaggle.com/c/imagenet-object-localization-challenge/data
[4] https://github.com/jingkang50/openood

the GLOW (Kingma & Dhariwal, 2018) architecture, as publicly available [5] under an MIT License. Additionally, inspired by recent advancements in typicality (Chali et al., 2023; Viviers et al., 2024), we incorporate the approximate mass-augmented log-likelihood objective in the model's training objective. Denoting the average log-likelihood (LL) of the model, parameterized by θ, evaluated over a batch of input data x as $L(x;\theta)$, the revised training objective is expressed as

$$\min_{\theta} \left(-L(x;\theta) + \alpha \left\| \frac{\partial L(x;\theta)}{\partial x} \right\| \right). \tag{6}$$

Here, $\alpha > 0$ serves as a hyperparameter governing the balance between local likelihood enhancement and gradient magnitude reduction, with $\alpha = 2$ employed in our GLOW implementation. For every dataset, we employed a GLOW architecture with 3 blocks of 32 affine coupling layers and 512 hidden units. All networks were trained using the Adam optimizer, with a learning rate of $5e^{-4}$. During testing, we compute the per-sample LL and gradient score.

A.2.2 DDPM-OOD

We implemented DDPM-OOD following the specifications outlined by Graham et al. (2023) and as publicly available [6] under an Apache 2.0 License. The method employs a time-conditioned UNet (Ronneberger et al., 2015) architecture with a simplified training objective where the variance is set to time-dependent constants and the model is trained to directly predict the noise ϵ at each timestep t, such that

$$L(\theta) = \mathbb{E}_{t,x_0,\epsilon} \left[\|\epsilon - \epsilon_\theta(x_t)\|^2 \right]. \tag{7}$$

The objective is to reconstruct an input x_t across multiple timesteps (t), employing the DDPM sampling strategy, which necessitates t steps for each reconstruction $\hat{x}_0, t$, with each step involving a model evaluation. To improve efficiency, we utilize the PLMS sampler (Liu et al., 2022), a recent advancement in rapid sampling for diffusion models. During the evaluation, we assess the reconstructions using both the mean-squared error (MSE) between the reconstructed and input images and the Learned Perceptual Image Patch Similarity (LPIPS) metric (Zhang et al., 2018). The latter evaluates perceptual similarity based on deep feature distances. For each of the N reconstructions, we compute these two similarity measurements. Subsequently, we average these scores to derive an OOD score for each input.

The model architecture is implemented exactly as described in (Graham et al., 2023). For training, we set $T = 1000$ and employed a linear noise schedule, with β_t ranging from 0.0015 to 0.0195. The training process spanned 300 epochs, utilizing the Adam optimizer with a learning rate of $2.5e^{-5}$. During the testing, we utilized the PLMS sampler configured to 100 timesteps and, in line with AnoDDPM (Wyatt et al., 2022), we only tested reconstructions from $T = 250$ for covariate shift. For the semantic shift experiments we utilize the full denoising schedule as investigated and suggested in (Graham et al., 2023).

A.2.3 MOODv2

For the implementation of MOODv2, we follow the guidelines provided in (Li et al., 2024) and made publicly available [7] under no official license. This work reduces the complexity of MOODv1 (Li et al., 2023a) while increasing the detection performance. In MOODv1, three steps were required: first, pre-train the Masked Image Modeling Vision Transformer (ViT) on the ImageNet-21k (Russakovsky et al., 2015) dataset; then the ViT must be fine-tuned on the same dataset; finally, the ViT is fine-tuned on the ID dataset. This becomes very expensive when dealing with a substantial number of ID datasets. However, through experimental validation, MOODv2 has demonstrated that a well-prepared masked image modeling model does not require additional fine-tuning.

The selected encoder is a BEiTv2 (Peng et al.), already pre-trained and fine-tuned on ImageNet-21k and provided in the aforementioned repository. Regarding the OOD score function, following the

[5] https://github.com/y0ast/Glow-PyTorch
[6] https://github.com/marksgraham/ddpm-ood
[7] https://github.com/dvlab-research/MOOD

author's recommendations, ViM (Wang et al., 2022) is utilized, which merges features and logits extracted from the trained image encoder. Here, l_i represents the i-th logit of feature x in the training set X; α denotes a model-specific constant; R (with dimensions $N \times (N - D)$) corresponds to the portion of the eigenvector matrix Q of X, ranging from the $(D + 1)$-th column to the last, where N stands for the principal dimension; and C signifies the number of classes. Mathematically, the score can be expressed as

$$s(x) = \frac{e^{\alpha\sqrt{x^T R R^T x}}}{\sum_{i=1}^{C} e^{l_i} + e^{\alpha\sqrt{x^T R R^T x}}}. \tag{8}$$

A.2.4 NNGUIDE

For the implementation of NNGuide, the guidelines in (Park et al., 2023) were followed, as well as the publicly available code [8] under the Apache 2.0 License. NNGuide is a post-hoc, training-free inference method designed to improve classifier-based OOD detection scores by leveraging nearest neighbors in the ID dataset. This method aims to mitigate the overconfidence issue in Far-OOD samples while preserving fine-grained detection for Near-OOD instances. This is achieved by augmenting a classifier's confidence score, $S_{\text{base}}(x)$, using a guidance term $G(x)$, which is the confidence-weighted average similarity of the nearest neighbors, ensuring that the score respects the data manifold's boundary geometry. The guidance term is defined as the average similarity between the test input and its high-confidence nearest neighbors and can be formulated as

$$S_{\text{NNGuide}}(x) = S_{\text{base}}(x) \cdot G(x). \tag{9}$$

A.2.5 SCALE

For the implementation of SCALE, the approach described by Xu et al. (2023) was followed, with the code made publicly available [9] under the MIT License. SCALE is a post-hoc enhancement method for OOD detection focusing on scaling network activations, rather than pruning them, using a simple but effective technique. The scaling factor r, derived from the activations, is applied uniformly across all features, preserving the model's logit ordinality and maintaining ID accuracy. Mathematically, the calculation of the logits with the scaled activations can be formulated as

$$z' = W \cdot (a \circ sf(a)) + b, \text{ where } sf(a)_j = \exp(r). \tag{10}$$

Furthermore, SCALE incorporates a training-time enhancement technique called Intermediate Tensor Shaping (ISH). ISH applies the same scaling concept during training to enhance OOD detection by emphasizing samples with stronger ID characteristics.

A.2.6 DC-GAN

To implement a DC-GAN, we comply with the architecture guidelines defined in (Radford et al., 2015). No pooling layers were used, instead strided convolutions are present in the discriminator, and fractional-strided convolutions in the generator. Additionally, batch normalization is incorporated into the generator and discriminator. Furthermore, we employ `ReLU` activation in the generator for all layers except the output layer, which utilizes `Tanh` activation. In contrast, we employ `LeakyReLU` activation in the discriminator for all layers. The code used is based on a publicly unlicensed available repository [10], although our implementation is provided in DisCoNet's repository.

For MNIST and CIFAR-10 datasets, with a resolution of 32×32 px, both the generator and the discriminator are constructed with 4 layers. In contrast, the ImageNet-200 dataset, with a resolution of 64×64 px, utilizes models with 5 layers. Throughout the architecture, the number of filters halves after each layer for the generator, while doubling for the discriminator. Both the generator and discriminator are optimized using the Adam optimizer, with β_1 set to 0.5 and β_2 set to 0.999. The discriminator learning rate lr_D and the generator learning rate lr_G for each ID dataset is specified in Table 8, along with additional implementation details.

[8] https://github.com/roomo7time/nnguide
[9] https://github.com/kai422/SCALE
[10] https://github.com/TeeyoHuang/conditional-GAN

Table 8: Hyperparameters used for DC-GAN's training.

ID Dataset	Latent Dimension	Generator Filters	Discriminator Filters	$\mathbf{lr_D}$	$\mathbf{lr_G}$
CIFAR-10	1024	256, 128, 64, 3	64, 128, 256, 1	$2e^{-4}$	$2e^{-4}$
ImageNet-200	1024	512, 256, 128, 64, 3	64, 128, 256, 512, 1	$2e^{-4}$	$2e^{-4}$

At test time, only the discriminator is utilized through a forward pass across the network to determine the probability of a sample belonging to the ID set. To calculate the anomaly score we follow Equation 5.

A.2.7 PRESCRIBED GAN

To implement PresGAN, we follow the recommendations presented by Dieng et al. (2019) and the official code repository [11], which does not state a license. An adapted version of this model is provided in DisCoNet's repository. PresGAN tackles two main limitations of the DC-GAN: mode collapse, which causes GANs to learn distributions with low support; and the lack of a probability density, making it impossible to evaluate generalization using predictive log-likelihood. PresGANs introduce noise to the output of a density network and optimize an entropy-regularized adversarial loss. This noise enables tractable approximations of the predictive log-likelihood and enhances training stability. The entropy regularizer encourages PresGANs to capture all modes of the data distribution. Fitting PresGANs involves computing intractable gradients of the entropy regularization term, which is addressed by using unbiased stochastic estimates. With θ and ϕ representing the Generator and Discriminator parameters, $\mathcal{L}_{GAN}(\theta, \phi)$ the generator's adversarial loss, λ the hyperparameter that controls the strength of the entropy regularization, and $p_\theta(x)$ the generative distribution induced by the generative process, the generator's training objective can be described by

$$\mathcal{L}_{PresGAN}(\theta, \phi) = \mathcal{L}_{GAN}(\theta, \phi) + \lambda \mathbb{E}_{p_\theta(x)} \left[\log p_\theta(x)\right]. \tag{11}$$

The generator and discriminator are similar to the ones used in the DC-GAN described in Appendix A.2.6 and summarized in Table 8. The latent dimension, generator filters, and discriminator filters are parameters shared by both. However, PresGAN requires the variance to be learned. To stabilize training and avoid failure cases, the variance σ of the generative distribution is truncated. Following the authors' recommendations, we defined the hyperparameters σ_{max} and σ_{min} that truncate the variance, the entropy regularization strength λ, and the generator learning rate, the discriminator learning rate and the sigma learning rate as shown in Table 9. The generator, discriminator, and σ are optimized using the Adam optimizer, with β_1 set to 0.5 and β_2 set to 0.999.

Table 9: Hyperparameters used for PresGAN's training.

ID Dataset	$\sigma_{\max}$	$\sigma_{\min}$	λ	$\mathbf{lr_D}$	$\mathbf{lr_G}$	$\mathbf{lr}_\sigma$
CIFAR-10	0.3	$1e^{-3}$	$5e^{-4}$	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$
ImageNet-200	0.3	$1e^{-3}$	$5e^{-4}$	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$

A.2.8 DISCONET

DisCoNet is an Adversarial VAE, comprised of a VAE and a Discriminator. The VAE features an Encoder ($\mathcal{E}_{\theta_E}$), consisting of convolutional layers with a kernel size of 3, stride 2, padding 1, and output padding of 1, followed by Batch Normalization and a `LeakyReLU` activation function. The number of filters doubles with each layer. Encoded features are then flattened and passed through two distinct fully connected layers, one estimating z_μ and the other z_σ, with outputs the size of the latent dimension. These outputs undergo the reparametrization trick to generate z, which is then fed into the VAE's decoder, referred to as the Generator ($\mathcal{G}_{\theta_G}$). The Generator comprises transposed convolutions, followed by Batch Normalization and a `LeakyReLU` activation, with the same kernel size, stride, padding, and output padding as the Encoder. However, the number of filters halves

[11] https://github.com/adjidieng/PresGANs

after each layer. A final convolutional layer with a kernel size of 3 and padding of 1, followed by a `Tanh` activation, generates the final output image. The generated image is subsequently fed into a Discriminator ($\mathcal{D}_\phi$). The Discriminator shares the same architecture as the Encoder but replaces the two fully connected layers with a single one that generates an output of size 1, followed by a `Sigmoid` activation. The training process of DisCoNet is covered in detail in Subsection 3.2, but can be summarized by Algorithm 1.

Algorithm 1 Training algorithm of DisCoNet.

Initialize parameters of models θ, ϕ
while training **do**
$x^{real} \leftarrow$ batch of images from dataset
$z_\mu^{real}, z_\sigma^{real} \leftarrow \mathcal{E}_{\theta_E}(x^{real})$
$z^{real} \leftarrow z_\mu^{real} + \epsilon_{real} z_\sigma^{real}$ with $\epsilon_{real} \sim \mathcal{N}(0, \mathbf{I})$
$x^{rec} \leftarrow \mathcal{G}_{\theta_G}(z^{real})$
$z^{fake} \leftarrow \epsilon_{fake}$ with $\epsilon_{fake} \sim \mathcal{N}(0, \mathbf{I})$
$x^{fake} \leftarrow \mathcal{G}_{\theta_G}(z^{fake})$
$x^{rec} \leftarrow \mathcal{G}_{\theta_G}(z^{real})$
$D^{real} \leftarrow \mathcal{D}_\phi(x^{real})$
$D^{rec}, D^{fake} \leftarrow \mathcal{D}_\phi(x^{rec}), \mathcal{D}_\phi(x^{fake})$

$\theta \leftarrow \nabla_\theta \mathcal{L}_{\text{VAE}}(\theta)$
$\phi \leftarrow \nabla_\phi \mathcal{L}_{\text{D}}(\phi)$
end while

The VAE and the Discriminator are optimized using the Adam optimizer, with β_1 set to 0.9 and β_2 set to 0.999. Both models share the same learning rate, represented by lr. As demonstrated in Equation4, three weighing terms are required to train the model: ω_{KL}, ω_{Rec} and ω_{Gen}. The weighing terms were fixed for all datasets, with $\omega_{\text{KL}} = 1e^{-4}$, $\omega_{\text{Rec}} = 1e^{-3}$ and $\omega_{\text{Gen}} = 1e^{-3}$. For each of the ID datasets, the hyperparameters used for training the DisCoNet can be found in Table 10.

Table 10: Hyperparameters used for DisCoNet's training.

ID Dataset	Latent Dimension	Encoder Filters	Generator Filters	lr
CIFAR-10	1024	64, 128, 256, 512	512, 256, 128, 64	$5e^{-4}$
ImageNet-200	1024	64, 128, 256, 512	512, 256, 128, 64	$5e^{-4}$
ImageNet-1K	1024	64, 128, 256, 512, 1024	1024, 512, 256, 128, 64	$1e^{-4}$

The developed code is based on a publicly available repository [12] under the Apache 2.0 License.

A.3 Compute Resources

This appendix describes the computational resources required to train the selected models on each dataset. Furthermore, each model's inference time is reported, allowing for an assessment of its suitability for real-world applications.

A.3.1 Training

The adversarial models were trained on a system featuring an NVIDIA TITAN Xp GPU with 12 GB VRAM, paired with a 6-core, 12-thread AMD Ryzen 5500 CPU and 16 GB RAM, referred to as System A. MOODv2, NNGuide, and SCALE did not require additional training. GLOW and DDPM-OOD models were trained on a system with an NVIDIA TITAN RTX GPU (24 GB VRAM), a 24-core, 48-thread AMD EPYC 7402P CPU, and 48 GB RAM, referred to as System B. DisCoNet was trained on ImageNet-1K using a system equipped with an NVIDIA H100 Tensor Core GPU (94 GB VRAM), a 32-core, 64-thread AMD EPYC 9334 CPU, and 768 GB RAM, referred to as System C. More information can be found in Table 11.

[12] https://github.com/AntixK/PyTorch-VAE

Table 11: Summary of the compute resources required for training.

ID Dataset	Model	Batch Size	Epochs	Trainable Parameters	Total Time (s)	System
CIFAR-10	GLOW (LL)	32	250	44,235,312	166,912	B
	GLOW (Typ.)	32	250	44,235,312	166,912	B
	DDPM-OOD	512	300	17,714,563	10,531	B
	DC-GAN	512	200	5,516,928	2,254	A
	PresGAN	512	200	5,517,952	13,837	A
	DisCoNet	512	250	10,993,861	5,015	A
ImageNet-200	GLOW (LL)	12	250	44,235,312	401,825	B
	GLOW (Typ.)	12	250	44,235,312	401,825	B
	DDPM-OOD	128	300	17,714,563	74,772	B
	DC-GAN	512	200	13,911,680	19,846	A
	PresGAN	512	100	13,915,776	40,637	A
	DisCoNet	512	140	29,886,661	15,044	A
ImageNet-1K	DisCoNet	1,256	140	69,240,517	111,471	C

A.3.2 Inference

To determine the expected time it takes for the selected models to evaluate each image, we tested them all on System A. Each model had a fixed batch size of 512 and processed 10 batches, except for GLOW, which needs to compute one image at a time, and DDPM-OOD on ImageNet-200 due to VRAM limitations. The time it takes to process each batch was measured from when the images were fed into the model to when the anomaly scores were determined. Table 12 displays the average inference times measured *per batch* during the evaluation.

Table 12: Inference times of the tested models. Legend: *batch size of 1; †batch size of 128.

ID Dataset	Model	Model Parameters	Inference Time (ms)
CIFAR-10	GLOW (LL)	44,235,312	33,689.6*
	GLOW (Typ.)	44,235,312	91.289,6*
	DDPM-OOD	17,714,563	85,525.0
	MOODv2	86,530,984	5,117.9
	DC-GAN	663,296	3.8
	PresGAN	663,296	3.5
	DisCoNet	1,556,994	3.6
ImageNet-200	GLOW (LL)	44,235,312	36,711.7*
	GLOW (Typ.)	44,235,312	99,121.7*
	DDPM-OOD	17,714,563	170,334.8†
	MOODv2	86,530,984	5,148.5
	DC-GAN	2,765,568	16.6
	PresGAN	2,765,568	16.3
	DisCoNet	1,569,282	10.9
ImageNet-1K	MOODv2	86,530,984	5,179.3
	NNGuide	83,590,140	2,896.8
	SCALE	25,557,032	963.5
	DisCoNet	6,307,330	77.3

A.4 Detailed Covariate Shift Results on CIFAR-10

This appendix contains the performance metrics per corruption achieved on the CIFAR-10 Covariate Shift OOD benchmark for every evaluated model.

A.4.1 GLOW with Log-Likelihood

Table 13 results indicate that, when trained using the Log-Likelihood objective, GLOW is very sensitive to noise-related corruptions, achieving a good separation between ID and OOD samples. However, for high-frequency dampening perturbations, such as blurring, the performance decreases drastically.

Table 13: Covariate shift OOD benchmark for GLOW with log-likelihood trained on CIFAR-10.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	57.9/93.6	63.6/91.3	67.7/89.8	71.2/88.7	73.6/88.2	66.8/90.3
Contrast	39.2/95.6	20.9/98.6	14.9/98.7	8.8/99.3	2.2/99.8	17.2/98.4
Defocus Blur	44.2/95.7	31.8/97.0	21.0/98.1	17.7/98.6	9.3/99.0	24.8/97.7
Elastic Transform	38.9/95.8	34.2/96.8	27.8/97.1	36.2/96.5	51.5/93.9	37.7/96.0
Fog	44.7/95.0	33.7/96.1	27.8/96.6	24.2/96.5	22.1/95.2	30.5/95.9
Frost	74.0/78.0	79.5/79.0	82.9/66.1	83.2/60.2	84.3/48.2	80.8/66.3
Gaussian Blur	44.3/95.7	21.3/98.0	14.2/98.6	9.8/99.0	5.2/99.3	19.0/98.1
Gaussian Noise	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Glass Blur	87.8/65.4	84.7/77.3	79.3/83.9	87.6/69.9	82.8/78.1	84.4/74.9
Impulse Noise	99.5/2.4	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	99.9/0.5
JPEG Compression	49.3/90.9	44.2/93.8	42.0/94.5	39.9/95.7	36.2/96.2	42.3/94.2
Motion Blur	34.2/96.5	26.8/97.1	21.8/97.7	21.7/97.5	18.2/98.4	24.5/97.4
Pixelate	57.1/93.6	62.0/92.3	63.5/91.4	67.2/90.3	67.0/90.8	63.4/91.7
Saturate	23.9/97.4	12.1/98.7	69.4/88.2	88.0/52.1	92.2/42.8	57.1/75.8
Shot Noise	99.9/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Snow	63.0/88.6	75.0/80.0	72.0/81.7	70.9/87.0	71.1/90.9	70.4/85.6
Spatter	68.0/84.3	81.1/64.5	88.6/39.6	75.2/70.5	88.4/39.6	80.3/59.7
Speckle Noise	99.7/0.1	99.9/0.0	99.9/0.0	99.9/0.0	100.0/0.0	99.9/0.0
Zoom Blur	27.7/97.1	21.1/97.8	17.2/98.6	14.4/98.6	11.5/98.6	18.4/98.1
Average	60.7/71.9	57.5/71.5	58.4/69.5	58.7/68.4	58.7/66.3	58.9/69.5

A.4.2 GLOW WITH TYPICALITY

The results in Table 14 significantly differ from the ones shown by the model trained using Log-Likelihood. GLOW trained with Typicality performs significantly worse for the detection of noisy corruptions. Its best OOD detection scores occur for blur detection, although performance is limited.

Table 14: Covariate shift OOD benchmark for GLOW with typicality trained on CIFAR-10.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	49.3/95.8	44.7/96.7	39.6/97.0	35.2/97.7	29.0/98.6	39.6/97.2
Contrast	60.4/90.6	66.1/81.4	71.5/73.0	77.7/62.3	91.2/30.3	73.4/67.5
Defocus Blur	55.7/92.0	60.2/83.9	65.6/73.1	72.4/59.5	80.1/45.2	66.8/70.7
Elastic Transform	53.6/89.7	55.9/86.1	59.3/79.9	48.9/86.7	34.6/94.0	50.5/87.3
Fog	58.9/92.4	62.5/88.3	65.0/83.6	66.1/80.0	69.1/74.1	64.3/83.7
Frost	36.1/97.3	29.7/98.7	25.6/99.8	26.3/99.5	25.9/99.7	28.7/99.0
Gaussian Blur	55.4/92.9	65.1/74.9	71.1/62.5	76.5/53.4	85.5/36.6	70.7/64.1
Gaussian Noise	0.0/100.0	0.0/100.0	0.2/100.0	0.2/100.0	0.5/100.0	0.2/100.0
Glass Blur	14.8/100.0	17.2/99.8	19.8/99.0	14.8/100.0	16.8/99.5	16.7/99.7
Impulse Noise	13.3/99.1	11.0/100.0	12.4/100.0	19.3/100.0	29.3/99.8	17.1/99.8
JPEG Compression	83.1/54.0	84.5/47.8	85.0/47.2	84.9/44.4	84.1/47.0	84.3/48.1
Motion Blur	59.2/86.5	62.8/80.7	66.1/78.2	66.1/76.0	68.9/71.4	64.6/78.6
Pixelate	41.6/96.1	36.2/96.8	33.2/96.8	29.0/97.3	27.0/97.3	33.4/96.9
Saturate	71.8/68.0	92.6/24.3	45.3/97.6	17.6/99.8	9.4/100.0	47.3/77.9
Shot Noise	0.5/100.0	0.4/100.0	0.3/100.0	0.4/100.0	0.7/100.0	0.5/100.0
Snow	46.3/94.8	40.1/96.3	40.6/96.2	39.7/95.8	40.0/96.2	41.3/95.9
Spatter	33.3/96.8	20.2/97.6	12.6/98.0	26.4/97.3	14.0/98.3	21.3/97.6
Speckle Noise	0.7/100.0	0.6/100.0	0.6/100.0	0.8/100.0	1.8/100.0	0.9/100.0
Zoom Blur	62.1/80.0	66.2/72.1	69.2/66.9	72.0/61.1	75.6/54.7	69.0/67.0
Average	41.9/90.8	42.9/85.5	41.2/86.8	40.7/84.8	41.2/81.2	41.6/85.8

A.4.3 DDPM-OOD

Table 15 shows that, like the GLOW Log-Likelihood model, the DDPM-OOD model is more sensitive to noise corruptions. Nonetheless, the decrease in performance for other corruption types is less pronounced than for both GLOW models. The scores are determined using T=20 and LPIPS + MSE.

Table 15: Covariate shift OOD benchmark for DDPM-OOD trained on CIFAR-10.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	52.4/94.9	51.1/95.6	52.1/95.9	51.8/96.1	50.7/97.5	51.6/96.0
Contrast	48.2/95.3	48.4/95.3	48.2/95.6	46.0/95.9	44.7/95.9	47.1/95.6
Defocus Blur	48.7/94.8	49.7/95.5	57.2/92.8	69.9/89.9	91.1/51.0	63.3/84.8
Elastic Transform	51.9/94.0	51.7/93.9	55.4/94.0	52.4/93.1	49.9/93.6	52.3/93.7
Fog	54.4/92.3	62.5/86.0	69.2/78.1	76.7/73.7	85.6/54.2	69.7/76.9
Frost	52.0/96.5	54.5/97.3	61.5/95.3	64.1/95.1	70.5/86.4	60.5/94.1
Gaussian Blur	48.6/95.3	58.1/93.5	70.0/88.0	81.7/78.0	94.9/32.2	70.6/77.4
Gaussian Noise	75.9/78.1	92.0/30.3	98.3/4.8	99.2/2.1	99.6/1.1	93.0/23.3
Glass Blur	75.8/79.2	73.6/80.6	64.9/86.5	79.4/76.6	71.9/81.0	73.1/80.8
Impulse Noise	88.9/41.4	98.4/4.8	99.7/0.9	100.0/0.0	100.0/0.0	97.4/9.4
JPEG Compression	54.5/93.9	55.7/92.4	56.4/92.7	56.3/91.5	58.8/90.2	56.3/92.1
Motion Blur	54.4/93.6	63.5/91.2	71.9/86.3	72.1/88.8	79.1/82.1	68.2/88.4
Pixelate	50.4/95.3	54.1/94.0	53.3/93.5	57.4/91.3	61.8/91.2	55.4/93.1
Saturation	53.5/95.9	60.7/94.9	51.3/95.3	55.9/92.2	61.5/91.4	56.6/93.9
Shot Noise	67.6/85.5	79.9/64.6	95.8/15.8	97.8/7.3	99.3/2.1	88.1/35.1
Snow	58.6/93.6	67.3/89.2	64.3/91.3	60.1/94.4	57.0/96.7	61.5/93.0
Spatter	57.6/91.8	67.2/83.3	73.3/79.3	64.2/84.1	77.1/68.0	67.9/81.3
Speckle Noise	68.4/79.4	68.4/79.4	93.6/25.1	97.9/7.6	99.2/2.2	85.5/38.7
Zoom Blur	61.7/91.4	64.4/90.2	69.1/85.8	72.6/84.5	78.3/80.0	69.2/86.4
Average	59.1/88.5	64.3/81.7	68.7/73.5	71.3/70.6	75.3/63.0	67.8/75.5

A.4.4 MOODv2

Table 16 shows MOODv2's well-balanced performance across all corruption types, with no discernible trends indicating which are easier to detect. Furthermore, a direct correlation exists between detection performance and corruption intensity.

Table 16: Covariate shift OOD benchmark for MOODv2 trained on CIFAR-10.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	65.8/87.9	66.3/87.7	67.1/87.3	68.1/86.5	70.8/83.5	67.6/86.6
Contrast	64.8/88.6	64.6/88.7	65.0/88.5	66.2/87.9	72.6/83.1	66.6/87.4
Defocus Blur	66.9/86.9	69.8/84.8	72.2/83.0	74.3/82.3	77.4/80.9	72.1/83.6
Elastic Transform	78.3/73.2	76.3/77.0	77.1/78.2	82.2/71.4	86.1/64.3	80.0/72.8
Fog	66.4/87.9	70.4/85.5	74.7/80.7	79.8/75.3	85.8/66.7	75.4/79.2
Frost	68.4/86.7	71.0/84.5	73.8/82.3	74.3/81.5	76.8/79.2	72.9/82.9
Gaussian Blur	67.1/86.7	73.4/81.9	75.2/80.8	76.5/80.0	79.1/78.5	74.2/81.6
Gaussian Noise	79.1/73.0	83.2/64.1	86.8/55.7	88.2/51.8	89.6/48.0	85.4/58.5
Glass Blur	86.0/65.6	86.7/63.2	87.0/62.8	89.9/57.4	90.1/56.2	88.0/61.0
Impulse Noise	74.0/81.8	76.6/77.6	78.3/74.0	81.1/69.7	83.8/65.8	78.8/73.8
JPEG Compression	72.7/84.3	76.3/81.0	77.5/79.5	78.7/78.2	80.6/76.0	77.2/79.8
Motion Blur	73.4/81.8	77.8/77.3	81.9/71.8	81.6/72.1	84.7/67.6	79.9/74.1
Pixelate	66.8/87.2	70.7/84.1	70.6/84.3	76.0/78.8	86.0/66.5	74.0/80.2
Saturate	63.5/89.6	64.6/89.4	67.6/86.6	71.8/83.3	76.0/79.0	68.7/85.6
Shot Noise	75.4/79.0	78.5/73.6	83.9/62.4	85.9/57.6	88.6/51.3	82.5/64.8
Snow	75.4/79.2	81.7/71.4	81.2/71.3	81.1/71.8	82.3/69.6	80.3/72.7
Spatter	75.4/80.4	81.1/72.6	83.0/68.7	77.0/79.1	81.0/71.3	79.5/74.5
Speckle Noise	73.6/80.5	78.3/73.7	80.5/70.0	84.4/61.8	87.7/54.1	80.9/68.0
Zoom Blur	75.8/79.3	75.3/79.7	76.8/77.7	77.8/76.3	80.2/72.4	77.2/77.1
Average	72.0/82.1	74.9/78.8	76.9/76.1	78.7/73.8	82.1/69.2	76.9/76.0

A.4.5 DC-GAN

The DC-GAN, as demonstrated in Table 17, does not achieve good performance for Covariate Shift OOD detection. The best results it achieves concern the detection of noisy corruptions, but with a very limited ceiling when compared to the other models.

Table 17: Covariate shift OOD benchmark for the DC-GAN trained on CIFAR-10.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	47.9/96.0	46.1/96.8	44.4/97.3	43.1/97.6	41.6/98.0	44.6/97.1
Contrast	56.5/87.8	61.5/78.1	65.3/72.9	72.1/64.7	82.7/59.0	67.6/72.5
Defocus Blur	51.1/94.3	52.2/93.3	52.5/92.8	52.7/92.6	52.2/91.9	52.2/93.0
Elastic Transform	54.4/91.9	54.4/91.8	54.3/91.4	55.6/90.1	57.4/89.0	55.2/90.9
Fog	55.4/89.4	57.5/83.5	57.5/81.8	56.6/80.6	55.8/80.0	56.6/83.1
Frost	47.1/95.4	46.0/95.6	47.5/94.4	49.1/93.5	51.6/91.4	48.2/94.1
Gaussian Blur	51.1/94.4	52.6/93.0	52.6/92.7	52.4/92.4	52.0/92.0	52.1/92.9
Gaussian Noise	59.2/91.0	64.9/88.8	68.1/86.6	68.3/87.4	68.5/87.8	65.8/88.3
Glass Blur	59.6/88.5	60.1/88.2	61.4/86.7	60.2/87.4	62.9/84.3	60.8/87.0
Impulse Noise	44.0/95.9	46.3/96.5	50.0/96.0	53.5/96.1	53.5/97.0	49.5/96.3
JPEG Compression	52.3/93.4	53.2/92.7	53.5/92.4	53.9/92.1	54.1/91.6	53.4/92.4
Motion Blur	52.8/92.8	53.7/91.7	54.3/90.5	54.4/90.4	54.9/89.7	54.0/91.0
Pixelate	52.5/94.2	54.3/93.3	54.1/93.2	58.4/91.0	60.8/89.3	56.0/92.2
Saturate	51.4/90.7	47.8/91.7	44.7/98.1	40.3/99.4	38.6/99.7	44.5/95.9
Shot Noise	55.8/92.8	59.8/91.4	65.7/88.6	66.9/88.0	68.2/87.4	63.3/89.6
Snow	50.0/94.5	49.7/94.6	49.2/94.5	48.8/94.8	45.5/95.4	48.6/94.7
Spatter	50.4/94.4	51.0/94.2	52.2/93.6	53.0/94.2	53.8/93.6	52.1/94.0
Speckle Noise	55.5/92.8	60.7/90.9	63.2/89.8	66.7/88.0	68.3/87.4	62.9/89.8
Zoom Blur	52.2/93.3	52.5/92.8	52.6/92.8	52.5/92.8	52.4/92.4	52.5/92.8
Average	52.6/92.8	53.9/91.5	54.9/90.8	55.7/90.2	56.6/89.3	54.7/90.9

A.4.6 PRESGAN

Table 18 reveals a large performance leap when compared to DC-GAN, indicating that the mode collapse mitigation is effective in helping define the ID set boundary. The best results still occur for the same corruption type. However, performance metrics consistently improve across the board.

Table 18: Covariate shift OOD benchmark for the Prescribed GAN trained on CIFAR-10.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	50.5/94.9	52.1/94.6	54.3/93.6	57.6/92.5	65.6/88.2	56.0/92.7
Contrast	57.4/91.3	68.0/85.6	72.5/83.4	73.4/84.7	70.8/92.2	68.4/87.4
Defocus Blur	52.0/94.0	56.3/92.3	60.4/90.2	64.0/87.8	71.1/80.7	60.8/89.0
Elastic Transform	57.6/91.3	59.2/91.3	63.4/89.1	68.8/83.6	75.3/74.4	64.8/85.9
Fog	57.6/91.2	70.3/82.9	78.0/72.2	84.9/58.9	91.9/37.1	76.6/68.5
Frost	62.8/89.1	71.7/82.6	80.0/68.6	79.7/70.9	83.9/60.2	75.6/74.3
Gaussian Blur	52.3/93.9	60.6/89.5	64.0/87.0	67.0/84.8	72.3/80.1	63.3/87.1
Gaussian Noise	77.7/74.9	91.6/40.4	97.6/11.5	98.7/5.3	99.3/2.3	93.0/26.9
Glass Blur	84.2/58.2	83.1/61.8	81.3/65.6	88.5/48.3	87.3/50.8	84.9/57.0
Impulse Noise	76.2/77.3	90.2/44.4	96.0/19.8	99.4/2.2	99.9/0.3	92.3/28.8
JPEG Compression	60.0/91.0	65.4/87.3	66.9/85.6	69.0/83.9	71.5/82.0	66.5/86.0
Motion Blur	58.4/91.3	64.8/87.5	67.9/84.3	68.4/84.0	70.8/82.5	66.0/85.9
Pixelate	56.4/92.4	60.3/90.9	62.5/89.2	71.2/80.8	77.4/72.9	65.5/85.3
Saturate	76.5/76.8	80.5/70.3	40.8/97.8	45.7/97.6	54.4/96.3	59.6/87.8
Shot Noise	69.1/86.6	80.3/72.0	94.1/30.9	96.7/16.3	98.7/5.3	87.8/42.2
Snow	63.3/89.6	77.3/74.2	79.0/72.5	84.4/61.2	86.6/56.7	78.1/70.8
Spatter	57.4/92.5	68.2/85.5	79.1/71.3	61.7/90.3	68.7/84.5	67.0/84.8
Speckle Noise	69.7/85.0	85.4/62.9	90.6/46.1	96.0/21.3	98.2/8.6	88.0/44.8
Zoom Blur	60.0/90.1	61.3/89.1	62.9/87.4	64.8/87.2	67.2/84.6	63.2/87.7
Average	63.1/86.9	70.9/78.2	73.2/70.8	75.8/65.4	79.5/60.0	72.5/72.3

A.4.7 DISCONET

DisCoNet's performance, as showcased by Table 19, is vastly superior to all the other evaluated models for this task. Although DisCoNet, similarly to most models, is better at detecting noise-related shifts, the detection performance it shows for all the evaluated types of anomalies is consistently high.

Table 19: Covariate shift OOD benchmark for the DisCoNet trained on CIFAR-10.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	55.9/91.5	72.4/76.0	87.7/46.9	96.2/18.2	99.8/0.6	82.4/46.6
Contrast	93.0/31.9	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	98.6/6.4
Defocus Blur	79.8/67.9	100.0/0.1	100.0/0.0	100.0/0.0	100.0/0.0	95.9/13.6
Elastic Transform	99.6/1.5	100.0/0.1	100.0/0.0	100.0/0.0	100.0/0.0	99.9/0.3
Fog	66.5/86.2	96.3/17.7	99.9/0.4	100.0/0.0	100.0/0.0	92.5/20.9
Frost	98.5/7.5	100.0/0.1	100.0/0.0	100.0/0.0	100.0/0.0	99.7/1.5
Gaussian Blur	78.4/71.3	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	95.7/14.3
Gaussian Noise	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Glass Blur	100.0/0.1	99.9/0.3	99.9/0.2	100.0/0.0	100.0/0.1	100.0/0.2
Impulse Noise	99.9/0.5	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.1
JPEG Compression	92.7/33.3	96.9/14.7	97.8/10.3	98.4/7.2	99.2/3.6	97.0/13.8
Motion Blur	100.0/0.1	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Pixelate	73.9/77.5	89.8/42.7	92.3/34.7	99.1/3.8	99.9/0.3	91.0/31.8
Saturation	91.6/31.1	95.7/16.3	46.2/96.2	74.2/64.9	88.3/34.1	79.2/48.5
Shot Noise	99.9/0.2	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Snow	93.9/26.6	99.7/1.5	99.6/2.3	99.8/0.8	100.0/0.0	98.6/6.2
Spatter	86.8/48.0	99.2/4.4	100.0/0.0	98.4/8.0	99.9/0.1	96.8/12.1
Speckle Noise	99.9/0.2	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Zoom Blur	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Average	90.0/30.3	97.4/9.2	96.0/10.1	98.2/5.4	99.3/2.0	96.2/11.4

A.5 Detailed Covariate Shift Results on ImageNet-200

This appendix contains the performance metrics per corruption achieved on the ImageNet-200 Covariate Shift OOD benchmark for every evaluated model.

A.5.1 GLOW with Log-Likelihood

Table 20 demonstrates that the model behaves similarly to CIFAR-10(-C). It can detect noisy corruptions, especially at levels larger than or equal to two. However, it constantly deems blurry corruptions to be more in-distribution than uncorrupted samples. This results in an extremely low average AUROC and performance that does not improve with the severity of the corruption.

Table 20: Covariate shift OOD benchmark for GLOW trained with log-likelihood on ImageNet-200.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	38.0/96.5	42.4/95.9	45.6/95.6	47.6/95.9	48.2/96.5	44.4/96.1
Contrast	5.8/99.8	3.0/99.8	1.1/99.9	0.2/100.0	0.0/100.0	2.0/99.9
Defocus Blur	17.3/98.1	14.1/98.2	9.9/99.3	5.1/99.7	4.0/99.8	10.1/99.0
Elastic Transform	23.8/98.0	22.1/98.0	18.5/98.1	19.1/98.1	21.5/97.8	21.0/98.0
Fog	18.7/98.1	12.9/99.3	9.2/99.6	5.7/99.8	4.0/99.8	10.1/99.3
Frost	41.9/93.7	43.4/92.1	42.8/91.5	43.7/90.2	44.6/88.6	43.3/91.2
Gaussian Noise	65.6/65.5	97.5/6.9	99.7/0.8	100.0/0.2	100.0/0.0	92.6/14.7
Glass Blur	50.2/93.8	23.9/97.5	15.4/98.1	11.7/98.2	6.5/99.5	21.5/97.4
Impulse Noise	69.2/64.7	94.3/16.7	99.7/0.5	100.0/0.1	100.0/0.0	92.6/16.4
JPEG Compression	17.4/98.1	17.5/98.2	14.2/98.8	12.6/99.4	9.1/99.7	14.2/98.8
Motion Blur	21.9/98.0	17.4/98.1	14.4/98.2	12.2/98.6	10.6/98.9	15.3/98.4
Pixelate	32.7/97.0	32.3/96.9	33.0/96.6	30.6/97.0	28.5/97.3	31.4/97.0
Shot Noise	66.1/71.7	89.0/43.9	96.6/15.0	98.4/3.0	99.3/0.8	89.9/26.9
Snow	42.6/93.6	54.4/88.7	45.0/93.8	40.7/96.0	36.3/97.3	43.8/93.9
Zoom Blur	16.1/98.1	12.2/98.5	10.4/98.9	8.7/99.4	7.4/99.5	11.0/98.9
Average	35.2/91.0	38.4/81.9	37.0/79.0	35.8/78.4	34.7/78.4	36.2/81.7

A.5.2 GLOW WITH TYPICALITY

The GLOW model trained with the Typicality objective behaves similarly to its CIFAR-10(-C) counterpart, as demonstrated in Table 21. This model is better at detecting shifts related to high-frequency dampening, such as Fog. On the other hand, corruptions that add high-frequency content, such as Impulse Noise, are considered to be more ID than real samples.

Table 21: Covariate shift OOD benchmark for GLOW trained with typicality on ImageNet-200.

Corruption	**Corruption Intensity**					**Average**
	1	**2**	**3**	**4**	**5**	
Brightness	43.6/96.0	37.4/97.8	33.1/98.6	29.8/99.0	28.1/99.1	34.4/98.1
Contrast	81.0/48.7	86.9/36.3	92.6/22.5	97.4/7.0	99.2/3.4	91.4/23.6
Defocus Blur	59.3/79.3	61.5/74.4	66.0/66.8	73.1/53.0	75.8/48.4	67.2/64.4
Elastic Transform	56.7/85.4	57.4/83.7	59.3/79.8	58.0/80.6	56.3/83.0	57.6/82.5
Fog	65.1/73.1	71.9/62.9	76.9/54.3	82.6/45.9	86.4/36.5	76.6/54.5
Frost	44.9/95.0	46.2/93.9	48.8/91.4	49.7/90.5	50.8/90.1	48.1/92.2
Gaussian Noise	27.1/97.4	4.3/99.2	0.5/99.8	0.2/100.0	0.0/100.0	6.4/99.3
Glass Blur	45.7/94.6	57.6/80.8	61.9/72.1	64.6/67.6	70.6/56.9	60.1/74.4
Impulse Noise	22.3/98.1	5.0/99.2	0.4/100.0	0.1/100.0	0.0/100.0	5.5/99.5
JPEG Compression	64.5/79.8	65.7/80.9	69.9/74.4	73.9/70.8	81.1/58.7	71.0/72.9
Motion Blur	56.6/83.9	59.2/79.1	61.2/74.4	63.1/71.6	64.7/69.4	61.0/75.7
Pixelate	52.9/90.2	53.7/89.1	53.0/89.4	55.5/84.7	60.4/78.5	55.1/86.4
Shot Noise	27.7/97.4	13.8/99.0	5.2/99.6	2.8/99.8	1.3/100.0	10.1/99.2
Snow	51.3/92.2	47.8/94.3	54.2/91.4	57.9/90.2	63.2/85.8	54.9/90.8
Zoom Blur	60.2/78.2	63.4/71.6	65.2/68.4	67.4/63.8	69.1/61.2	65.1/68.6
Average	50.6/86.0	48.8/82.8	49.9/78.9	51.7/75.0	53.8/71.4	51.0/78.8

A.5.3 DDPM-OOD

For the DDPM-OOD model trained on ImageNet-200, we observe a paradigm inversion, according to the results found in Table 22. This model is better at detecting shifts related to high-frequency dampening, such as Defocus Blur, than corruptions that increase the high-frequency content present in the data distribution. The scores are determined using T=20 and LPIPS + MSE.

Table 22: Covariate shift OOD benchmark for DDPM-OOD trained on ImageNet-200.

Corruption	**Corruption Intensity**					**Average**
	1	**2**	**3**	**4**	**5**	
Brightness	61.4/85.4	55.3/90.3	51.7/91.1	51.8/91.1	52.8/91.8	54.6/89.9
Contrast	83.4/57.5	87.2/46.9	91.4/38.5	95.8/18.4	96.9/15.6	90.9/35.4
Defocus Blur	84.2/58.1	90.7/39.2	96.2/17.7	99.3/1.3	99.8/0.9	94.0/23.4
Elastic Transform	75.7/73.9	76.0/73.6	82.0/61.1	80.3/66.1	76.3/71.0	78.1/69.1
Fog	81.9/60.5	90.9/32.8	95.5/19.5	98.6/5.9	99.3/2.2	93.2/24.2
Frost	52.5/88.5	54.7/89.4	61.6/85.1	65.2/80.9	68.6/79.0	60.5/84.6
Gaussian Noise	47.0/90.9	60.0/86.3	71.5/75.4	77.0/65.1	79.0/64.6	66.9/76.5
Glass Blur	63.1/82.2	74.9/73.7	81.2/62.0	86.8/47.6	95.6/19.1	80.3/56.9
Impulse Noise	55.6/87.0	63.2/82.4	73.9/70.5	78.8/60.8	79.6/60.2	70.2/72.2
JPEG Compression	70.1/78.9	67.3/81.8	73.9/78.3	72.5/76.0	76.9/70.5	72.1/77.1
Motion Blur	80.0/66.2	84.2/59.2	90.6/39.1	94.1/22.6	95.9/16.5	88.9/40.7
Pixelate	64.0/82.2	65.1/83.6	71.8/74.1	76.0/68.4	80.4/64.0	71.4/74.5
Shot Noise	50.8/91.9	54.4/89.9	63.4/83.6	69.7/80.2	74.9/72.6	62.6/83.6
Snow	55.4/87.0	59.5/84.1	52.9/88.9	50.9/88.1	54.3/87.6	54.6/87.1
Zoom Blur	88.1/42.1	92.5/31.0	95.3/19.7	96.3/15.6	97.7/11.0	94.0/23.9
Average	67.6/75.5	71.7/69.6	76.8/60.3	79.5/52.5	81.9/48.4	75.5/61.3

A.5.4 MOODV2

For the model evaluated in ImageNet-200, there is an evident tendency in the detection performance: the best results occur for corruptions that filter high-frequency components. The intensity of the

corruption plays an important role for MOODv2 in this dataset, which achieves very low scores for Intensity 1 in every corruption tested, as demonstrated in Table 23.

Table 23: Covariate shift OOD benchmark for MOODv2 trained on ImageNet-200.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	54.3/91.8	55.4/91.4	57.7/90.3	61.6/88.4	66.6/85.0	59.1/89.4
Contrast	58.5/92.0	61.0/91.8	65.6/91.1	75.4/89.5	84.8/86.7	69.1/90.2
Defocus Blur	62.9/86.3	65.6/84.8	69.5/83.1	80.7/71.4	85.3/61.9	72.8/77.5
Elastic Transform	66.3/83.1	64.5/85.2	67.2/82.5	72.4/77.3	80.6/67.2	70.2/79.1
Fog	56.7/92.4	60.4/91.8	64.7/90.7	72.7/85.5	80.6/79.1	67.0/87.9
Frost	56.8/90.8	58.8/90.6	60.5/90.7	62.1/90.6	63.9/90.4	60.4/90.6
Gaussian Noise	58.8/89.0	63.2/87.9	70.8/82.9	75.3/79.5	79.3/75.7	69.5/83.0
Glass Blur	66.2/86.8	74.2/75.4	82.2/63.7	86.3/61.7	87.2/64.2	79.2/70.3
Impulse Noise	60.5/89.6	63.5/86.7	70.0/82.8	74.6/80.6	81.8/75.7	70.1/83.1
JPEG Compression	57.2/92.0	56.4/93.2	58.3/92.0	59.7/91.7	64.9/88.8	59.3/91.5
Motion Blur	61.0/87.4	64.6/85.4	68.1/82.8	71.8/79.8	75.4/75.6	68.2/82.2
Pixelate	55.9/91.6	60.1/88.1	65.2/83.9	67.1/83.8	72.4/80.1	64.1/85.5
Shot Noise	57.7/90.2	60.8/88.7	66.4/85.3	71.5/81.4	79.1/75.0	67.1/84.1
Snow	59.4/90.1	64.5/88.0	67.3/85.4	73.0/80.4	72.1/81.9	67.3/85.2
Zoom Blur	65.4/84.0	69.4/79.9	73.5/74.9	78.7/66.4	83.0/57.5	74.0/72.5
Average	59.8/89.1	62.8/87.3	67.1/84.1	72.2/80.5	77.1/76.3	67.8/83.5

A.5.5 DC-GAN

Once again, the DC-GAN performance is subpar for the detection of Covariate Shift OOD samples. With the exception of the Snow corruption, as shown in Table 24, the model severely struggles to separate ID samples from OOD samples. Despite its limited detection capabilities, performance increases with the severity of the corruptions and it is more consistent than the one shown by GLOW in this task.

Table 24: Covariate shift OOD benchmark for the DC-GAN trained on ImageNet-200.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	54.3/92.6	55.1/94.0	57.8/94.2	61.8/93.8	66.8/92.5	59.2/93.4
Contrast	52.8/83.8	53.2/84.2	53.3/85.3	47.8/88.9	42.9/91.1	50.0/86.7
Defocus Blur	59.7/86.4	60.5/85.5	62.8/82.2	65.8/77.7	66.4/76.6	63.0/81.7
Elastic Transform	58.5/88.3	58.4/88.2	59.4/86.7	60.3/85.4	60.6/84.7	59.5/86.6
Fog	56.9/81.7	58.4/76.8	60.1/73.2	63.5/67.4	66.8/62.2	61.1/72.3
Frost	62.2/87.7	65.6/83.6	66.7/80.6	68.3/77.6	69.5/74.9	66.4/80.9
Gaussian Noise	56.8/89.2	55.2/88.8	54.5/90.0	55.0/90.7	55.5/91.2	55.4/90.0
Glass Blur	56.7/87.8	61.4/83.0	63.7/79.9	65.2/76.5	66.0/74.1	62.6/80.2
Impulse Noise	55.7/89.6	54.9/89.8	55.5/90.6	56.4/90.7	56.9/92.2	55.9/90.6
JPEG Compression	55.2/90.2	54.5/91.2	55.3/90.2	55.2/90.4	55.1/90.0	55.1/90.4
Motion Blur	58.5/88.0	60.2/85.9	61.5/84.2	62.5/83.0	63.3/81.2	61.2/84.5
Pixelate	55.5/90.4	57.8/88.7	56.4/88.9	57.8/88.0	58.8/87.0	57.3/88.6
Shot Noise	56.8/89.5	56.5/89.5	55.7/90.1	56.1/90.2	57.7/90.7	56.6/90.0
Snow	63.1/85.8	65.4/83.1	71.9/76.2	78.0/68.2	80.9/61.0	71.9/74.9
Zoom Blur	61.3/85.1	63.0/83.0	63.7/81.7	64.3/80.6	64.8/79.9	63.4/82.0
Average	57.6/87.7	58.7/86.4	59.9/84.9	61.2/83.3	62.1/81.9	59.9/84.8

A.5.6 PRESGAN

Table 25 shows that the PresGAN model trained on ImageNet-200 displays the same behavior as the one trained on CIFAR-10, i.e., the best results occur for the same type of corruptions. It also shows a significant improvement when compared to the DC-GAN, showing once more the positive effects of its strategy to achieve better mode coverage during training.

Table 25: Covariate shift OOD benchmark for the Prescribed GAN trained on ImageNet-200.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	50.5/93.4	48.1/94.0	48.7/94.2	49.0/94.4	48.5/94.7	49.0/94.1
Contrast	45.5/93.2	44.4/93.2	42.4/93.9	38.5/94.9	36.3/95.0	41.4/94.0
Defocus Blur	63.5/87.6	64.2/86.9	65.7/86.0	60.2/89.1	55.3/91.2	61.8/88.2
Elastic Transform	59.4/90.6	62.0/89.1	63.7/87.7	73.4/80.2	80.0/73.4	67.7/84.2
Fog	66.2/85.0	71.4/80.0	74.7/75.8	79.8/67.9	83.6/60.1	75.2/73.8
Frost	58.6/90.2	60.6/89.3	62.1/87.2	64.9/85.9	67.4/83.7	62.7/87.2
Gaussian Noise	61.9/89.0	76.5/78.4	88.7/53.6	93.9/32.5	96.8/15.8	83.6/53.8
Glass Blur	72.8/81.3	70.8/82.1	70.5/82.7	67.5/84.8	62.0/88.6	68.7/83.9
Impulse Noise	66.3/86.3	74.9/80.0	89.7/49.9	95.4/24.6	99.0/4.0	85.1/49.0
JPEG Compression	58.0/90.9	58.7/90.2	59.7/89.9	61.3/89.5	63.8/88.0	60.3/89.7
Motion Blur	62.3/88.6	63.5/87.7	65.2/86.3	65.7/86.0	65.9/85.7	64.5/86.8
Pixelate	61.3/89.3	64.1/87.8	66.9/86.2	71.2/83.3	68.9/84.8	66.5/86.3
Shot Noise	63.8/88.2	72.5/82.2	81.8/70.9	89.2/52.4	95.2/25.2	80.5/63.8
Snow	64.2/87.1	69.6/83.5	77.7/75.2	86.5/58.0	80.3/73.2	75.7/75.4
Zoom Blur	63.6/87.5	63.7/87.3	63.9/86.8	62.8/88.0	60.9/88.7	63.0/87.6
Average	61.2/88.5	64.3/86.1	68.1/80.4	70.6/74.1	70.9/70.1	67.0/79.9

A.5.7 DISCONET

As demonstrated in Table 26, DisCoNet excels at detecting every type of corruption at every available intensity. The performance gap observed when compared to the other models is pronounced for every evaluated scenario.

Table 26: Covariate shift OOD benchmark for the DisCoNet trained on ImageNet-200.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	99.6/1.7	98.3/10.2	98.1/12.6	98.8/7.4	99.3/2.4	98.8/6.9
Contrast	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Defocus Blur	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Elastic Transform	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Fog	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Frost	98.2/11.5	98.8/6.8	99.2/3.9	99.3/2.9	99.4/2.3	99.0/5.5
Gaussian Noise	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Glass Blur	99.6/1.6	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	99.9/0.3
Impulse Noise	99.9/0.1	99.9/0.2	99.9/0.1	100.0/0.0	100.0/0.0	99.9/0.1
JPEG Compression	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Motion Blur	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Pixelate	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Shot Noise	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Snow	97.8/13.3	98.7/8.0	97.9/13.9	98.5/9.5	98.7/7.2	98.3/10.4
Zoom Blur	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Average	99.7/1.9	99.7/1.7	99.7/2.0	99.8/1.3	99.8/0.8	99.7/1.5

A.6 DETAILED COVARIATE SHIFT RESULTS ON IMAGENET-1K

This appendix contains the performance metrics per corruption achieved on the ImageNet-1K Covariate Shift OOD benchmark for every evaluated model.

A.6.1 MOODV2

MOODv2 operates similarly in ImageNet-1K as it did in ImageNet-200; the best results are obtained for corruptions that filter high-frequency components. Table 27 shows that it scores very low for Intensity 1 in all corruption tests.

Table 27: Covariate shift OOD benchmark for MOODv2 evaluated on ImageNet-1K.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	54.2/93.8	55.2/93.2	56.9/92.2	59.2/90.6	62.2/88.2	57.5/91.6
Contrast	59.6/91.6	61.7/90.4	66.1/87.7	77.5/76.6	86.4/54.9	70.3/80.2
Defocus Blur	69.6/80.3	76.0/70.7	85.5/51.0	91.8/33.9	95.4/21.0	83.7/51.4
Elastic Transform	60.2/88.5	75.9/62.7	63.2/85.1	70.8/75.3	87.1/44.2	71.4/71.1
Fog	70.0/82.5	77.3/71.5	89.2/40.0	93.3/25.6	97.0/11.4	85.3/46.2
Frost	61.8/88.8	70.3/79.9	75.6/71.7	77.1/69.7	80.0/63.5	72.9/74.7
Glass Blur	60.6/90.6	72.1/78.0	81.6/62.0	88.9/44.4	96.5/17.6	79.9/58.5
Gaussian Blur	58.0/90.0	60.0/87.7	64.8/82.8	71.5/74.1	80.3/59.0	66.9/78.7
Gaussian Noise	63.2/86.3	70.2/77.8	83.9/51.0	87.9/41.4	93.3/26.5	79.7/56.6
Impulse Noise	57.2/89.8	60.4/86.8	63.6/83.5	71.0/74.6	78.9/61.6	66.2/79.3
JPEG Compression	63.5/88.0	65.9/85.7	67.6/83.8	71.9/77.6	77.5/68.5	69.2/80.7
Motion Blur	58.7/90.2	63.1/85.9	70.5/77.0	80.0/61.0	85.8/48.5	71.6/72.5
Pixelate	55.8/92.2	57.5/90.9	61.0/87.9	67.7/81.0	83.8/57.1	65.2/81.8
Saturate	54.0/93.4	55.6/92.2	55.4/93.1	60.3/90.0	65.2/85.2	58.1/90.8
Shot Noise	58.3/89.7	61.0/86.8	65.3/82.1	73.7/70.7	80.1/59.1	67.7/77.7
Snow	62.3/87.4	70.8/77.8	70.7/78.9	75.6/71.4	77.2/67.7	71.3/76.6
Spatter	55.3/92.9	59.1/90.2	62.0/87.8	64.2/85.7	69.4/80.0	62.0/87.3
Speckle Noise	57.6/90.4	59.1/88.8	63.9/83.7	67.4/79.3	72.3/72.0	64.1/82.8
Zoom Blur	65.8/83.7	71.9/75.5	76.7/67.1	81.3/58.3	86.5/46.2	76.5/66.1
Average	60.3/88.9	65.4/82.8	69.7/76.2	75.3/67.4	81.8/54.3	70.5/73.9

A.6.2 NNGUIDE

NNGuide surpasses the performance of MOODv2, as demonstrated by the results in Table 28, particularly for higher corruption intensities. Nonetheless, it also suffers from significantly low scores at Intensity 1 across all corruption tests.

Table 28: Covariate shift OOD benchmark for NNGuide evaluated on ImageNet-1K.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	56.5/90.0	58.8/88.5	62.7/85.6	68.3/80.5	74.7/72.6	64.2/83.4
Contrast	54.1/91.3	61.9/86.2	75.0/72.1	93.3/27.8	99.2/3.0	76.7/56.1
Defocus Blur	74.7/69.2	81.2/57.6	90.3/36.5	94.7/22.6	97.2/13.0	87.6/39.8
Elastic Transform	64.8/83.3	80.4/65.4	77.6/66.9	86.4/51.0	95.7/22.1	81.0/57.7
Fog	69.5/79.0	74.6/73.2	80.7/63.9	84.5/55.3	92.2/32.8	80.3/60.8
Frost	71.7/73.9	84.8/49.8	91.0/33.4	91.7/31.0	94.2/22.7	86.7/42.2
Gaussian Blur	50.3/93.5	65.9/81.8	77.6/67.9	85.9/51.7	94.7/23.6	74.9/63.7
Gaussian Noise	66.2/82.5	74.4/73.1	84.9/53.6	93.8/27.1	98.6/6.4	83.6/48.5
Glass Blur	76.6/68.1	86.4/47.7	96.1/16.5	97.6/10.5	98.6/6.2	91.1/29.8
Impulse Noise	78.1/67.9	82.3/61.0	85.9/52.5	93.8/27.6	98.3/8.2	87.7/43.5
JPEG Compression	63.4/84.7	66.6/81.5	69.3/78.4	77.6/66.5	87.2/46.5	72.8/71.5
Motion Blur	69.3/76.6	78.8/62.3	88.7/41.2	94.9/22.1	97.0/14.0	85.7/43.2
Pixelate	63.3/86.5	65.2/84.9	75.6/71.9	86.9/48.2	92.2/32.1	76.7/64.7
Saturate	47.3/96.3	49.9/95.2	45.5/96.9	57.2/94.6	66.5/91.2	53.3/94.8
Shot Noise	68.1/80.7	77.2/69.5	86.1/51.7	95.0/23.1	97.9/10.3	84.9/47.1
Snow	74.9/74.5	89.2/42.3	87.6/48.4	92.8/31.5	95.2/21.5	87.9/43.7
Spatter	43.5/97.1	55.6/95.7	65.0/93.6	71.2/91.4	78.2/87.1	62.7/93.0
Speckle Noise	52.0/95.1	57.2/93.2	71.5/84.2	78.6/75.7	85.4/62.5	68.9/82.2
Zoom Blur	77.2/68.8	83.4/57.8	87.6/47.8	90.5/39.5	92.9/31.3	86.3/49.0
Average	64.3/82.0	72.3/71.9	78.9/61.2	86.0/46.2	91.4/32.0	78.6/58.7

A.6.3 DISCONET

As seen in Table 29, DisCoNet excels at detecting every sort of corruption at each possible intensity on ImageNet-1K. This behavior is similar to what was observed in the other two datasets.

Table 29: Covariate shift OOD benchmark for DisCoNet trained on ImageNet-1K.

Corruption	Corruption Intensity					Average
	1	2	3	4	5	
Brightness	94.5/21.8	98.8/5.9	99.9/0.2	100.0/0.0	100.0/0.0	98.6/5.6
Contrast	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Defocus Blur	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Elastic transform	93.3/24.9	97.7/11.8	99.6/1.6	99.9/0.2	100.0/0.0	98.1/7.7
Fog	99.6/1.5	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	99.9/0.3
Frost	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Gaussian Blur	98.8/5.9	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	99.8/1.2
Gaussian Noise	99.8/0.9	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.2
Glass Blur	99.9/0.3	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.1
Impulse Noise	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.00/0.0
JPEG Compression	90.2/31.8	92.0/26.4	93.0/24.1	95.0/19.6	96.9/14.3	93.4/23.3
Motion Blur	99.4/2.6	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	99.9/0.5
Pixelate	95.4/19.3	96.3/16.7	98.9/5.4	99.9/0.4	100.0/0.1	98.1/8.4
Saturate	88.1/30.7	89.9/30.7	94.4/22.0	99.4/2.9	100.0/0.0	94.3/17.3
Shot Noise	99.5/2.2	100.0/0.1	100.0/0.0	100.0/0.0	100.0/0.0	99.9/0.5
Snow	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.00/0.0
Spatter	93.1/24.0	99.5/2.3	100.0/0.0	100.0/0.0	100.0/0.0	98.5/5.3
Speckle Noise	98.0/9.5	99.3/3.1	100.0/0.0	100.0/0.0	100.0/0.0	99.5/2.5
Zoom Blur	100.0/0.1	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0	100.0/0.0
Average	97.3/9.2	98.6/5.1	99.2/2.8	99.7/1.2	99.8/0.8	98.9/3.8

A.7 Frequency Impact of Corruptions

To assess the impact of corruptions on the frequency spectrum, we first calculated CIFAR-10's average Power Spectral Density (PSD) at its original resolution of 32×32 pixels. This involved computing the Fast Fourier Transform (FFT) on each channel of every image in the dataset. We aggregated all FFTs and, following Li et al. (2023b), determined the average Radial Profile of the dataset through Azimuthal averaging using a publicly available toolkit [13]. The frequency radius is the distance from the center of the FFT that represents the zero-frequency component; larger radii correspond to higher frequencies, with a maximum radius of 22 for this image size. We repeated this process for each corruption at maximum intensity (severity 5 per the benchmark), and the Relative Radial Profile of the PSD was calculated by dividing the Radial profile of each corrupted dataset by that of the uncorrupted dataset. Based on their effect on the frequency spectrum, it is possible to divide the corruptions into two categories: those that enhance high-frequency content and those that reduce it. Our analysis focuses on radii greater than 13, which represent higher frequency components, up to 21, since the final component contains only very residual information.

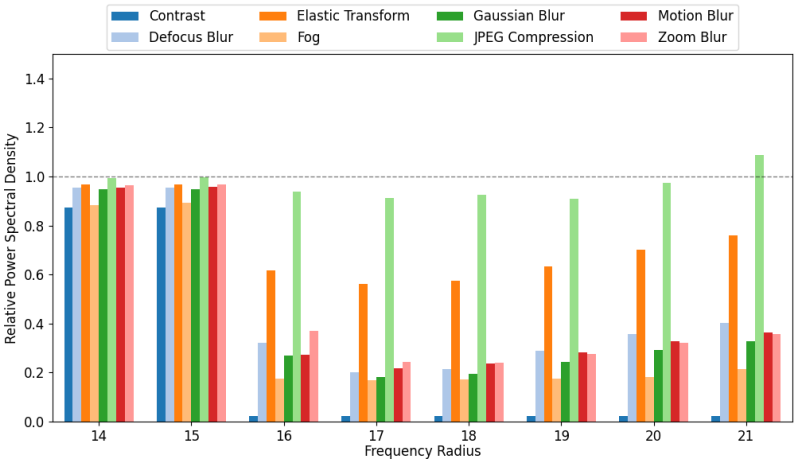

Figure 4: Relative radial profile of the PSD for the corruptions that dampen higher frequencies.

[13] https://github.com/keflavich/image_tools

As shown in Figure 4, this group of 8 corruptions leads to a dampening of the spectral components at higher frequency radius levels. The impact varies among the corruptions; for instance, the effect is quite pronounced in the case of Contrast, whereas it is more gradual and subtle in the case of JPEG Compression, approaching almost borderline levels.

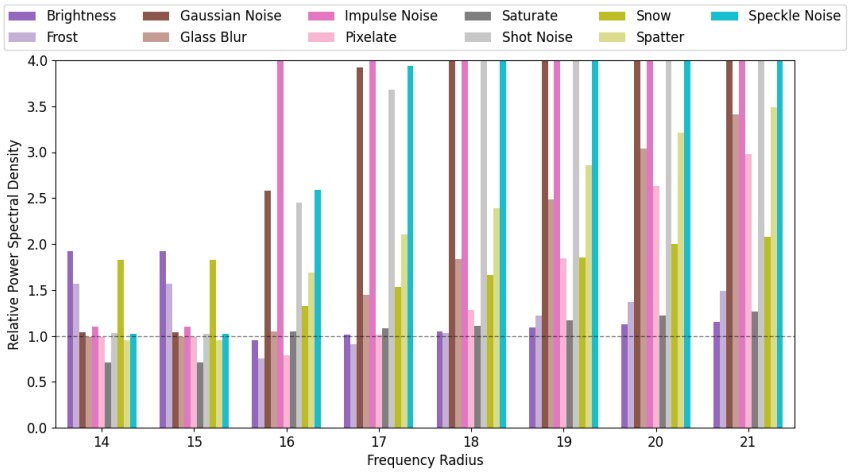

Figure 5: Relative radial profile of the PSD for the corruptions that amplify higher frequencies.

As Figure 5 demonstrates, this set of 11 corruptions results in an amplification of spectral components at higher frequency radius levels. The degree of this increase varies among the corruptions. For example, Gaussian Noise, Impulse Noise, and Speckle Noise exhibit a significant amplification of these components, while other corruptions like Brightness show a more moderate albeit consistent increase.

These outcomes were used to propose the splitting of the corruptions found in CIFAR-10(-C), ImageNet-200(-C) and ImageNet-1K(-C) into two distinct categories: **Lower Frequency** refers to corruptions that decrease higher frequencies, while **Higher Frequency** refers to those that increase high-frequency components. The proposed split for the corruptions is summarized in Table 30. It should be noted that four of the listed corruptions are not present in ImageNet-200(-C), mainly Gaussian Blur, Saturate, Spatter, and Speckle Noise.

Table 30: Corruptions categorized by frequency spectrum effects. Legend: *only present in CIFAR-10(-C).

Type	Corruptions
Higher Frequency	Brightness, Frost, Gaussian Noise, Glass Blur, Impulse Noise, Pixelate, Saturate*, Shot Noise, Snow, Spatter*, Speckle Noise*
Lower Frequency	Contrast, Defocus Blur, Elastic Transform, Fog, Gaussian Blur*, JPEG Compression, Motion Blur, Zoom Blur

A.8 DisCoNet Ablation Study

For the ablation study, we trained models using only reconstructions and only generated images, then analyzed their performance per type of corruption in terms of frequency, following the split proposed in Appendix A.7. The main goal was to demonstrate that the effectiveness of DisCoNet comes from this dual training strategy and not merely from the adversarial setting.

Table 31 demonstrates a clear pattern: models trained solely on reconstructions excel at detecting low-frequency corruptions, whereas models trained exclusively on generated images are more effective at recognizing high-frequency corruptions. However, for certain borderline cases identified in Appendix A.7, such as Brightness and Pixelate, this paradigm shows a slight shift.

Table 31: Ablation study showing the impact of using reconstructed or generated images during DisCoNet's training.

Corruption	CIFAR-10(-C)			ImageNet-200(-C)		
	Recon.	Generated	Both	Recon.	Generated	Both
Brightness	18.7/99.6	60.4/90.4	82.4/46.6	99.9/0.1	94.3/22.2	98.8/6.9
Contrast	99.2/3.9	78.3/58.1	98.6/6.4	100.0/0.0	99.8/0.3	100.0/0.0
Defocus Blur	98.6/6.4	74.0/68.3	95.9/13.6	100.0/0.0	99.7/0.2	100.0/0.0
Elastic Transform	99.9/0.5	69.9/79.6	99.9/0.3	100.0/0.0	99.0/3.9	100.0/0.0
Fog	89.3/31.1	75.5/62.6	92.5/20.9	100.0/0.0	98.2/8.1	100.0/0.0
Frost	46.0/99.6	82.6/59.1	99.7/1.5	99.9/0.2	99.8/0.1	99.0/5.5
Gaussian Blur	98.4/6.9	79.6/57.9	95.7/14.3	—	—	—
Gaussian Noise	8.8/99.6	97.3/11.6	100.0/0.0	90.2/0.0	99.9/0.1	100.0/0.0
Glass Blur	34.8/83.8	66.3/84.2	100.0/0.2	100.0/0.0	99.3/3.8	99.9/0.3
Impulse Noise	0.1/100.0	96.1/15.2	100.0/0.1	67.1/43.8	99.7/0.6	99.9/0.1
JPEG Compression	86.8/47.7	61.3/89.7	97.0/13.8	100.0/0.0	84.2/64.8	100.0/0.0
Motion Blur	100.0/0.0	86.0/50.1	100.0/0.0	100.0/0.0	99.1/3.2	100.0/0.0
Pixelate	78.5/62.9	68.9/81.1	91.0/31.8	100.0/0.0	91.8/38.0	100.0/0.0
Saturate	37.4/91.0	66.5/71.5	79.2/48.5	—	—	—
Shot Noise	13.6/99.3	95.2/18.9	100.0/0.0	86.1/19.3	99.8/0.2	100.0/0.0
Snow	14.3/100.0	82.0/65.3	98.6/6.2	99.1/5.0	97.6/9.7	98.3/10.4
Spatter	2.8/100.0	80.4/62.7	96.8/12.1	—	—	—
Speckle Noise	9.4/99.7	96.5/14.3	100.0/0.0	—	—	—
Zoom Blur	100.0/0.0	80.9/65.3	100.0/0.0	100.0/0.0	99.8/0.1	100.0/0.0
Lower Frequency	96.5/12.1	75.7/66.4	97.5/8.7	100.0/0.0	97.1/11.5	100.0/0.0
Higher Frequency	24.0/94.1	81.1/52.2	95.2/13.4	92.8/11.0	98.1/8.1	99.5/2.9
Average	54.6/59.6	78.8/58.2	96.2/11.4	96.2/5.8	97.6/9.7	99.7/1.5