

# Denoising Diffusion Models for Out-of-Distribution Detection

Mark S. Graham  
King's College London  
mark.graham@kcl.ac.uk

Walter H.L. Pinaya  
King's College London  
walter.diaz-sanz@kcl.ac.uk

Petru-Daniel Tudosiu  
King's College London  
petru.tudosiu@kcl.ac.uk

Parashkev Nachev  
University College London  
p.nachev@ucl.ac.uk

Sebastien Ourselin  
King's College London  
sebastien.ourselin@kcl.ac.uk

M. Jorge Cardoso  
King's College London  
m.jorge.cardoso@kcl.ac.uk

## Abstract

Out-of-distribution detection is crucial to the safe deployment of machine learning systems. Currently, the state-of-the-art in unsupervised out-of-distribution detection is dominated by generative-based approaches that make use of estimates of the likelihood or other measurements from a generative model. Reconstruction-based methods offer an alternative approach, in which a measure of reconstruction error is used to determine if a sample is out-of-distribution. However, reconstruction-based approaches are less favoured, as they require careful tuning of the model's information bottleneck - such as the size of the latent dimension - to produce good results. In this work, we exploit the view of denoising diffusion probabilistic models (DDPM) as denoising autoencoders where the bottleneck is controlled externally, by means of the amount of noise applied. We propose to use DDPMs to reconstruct an input that has been noised to a range of noise levels, and use the resulting multi-dimensional reconstruction error to classify out-of-distribution inputs. Our approach outperforms not only reconstruction-based methods, but also state-of-the-art generative-based approaches.

## 1. Introduction

Out-of-distribution (OOD) detection plays a crucial role in the safe deployment of machine learning systems, ensuring that downstream models are only run on data sampled from the distribution they were trained on. OOD detection models can be broadly divided into unsupervised models, which only require in-distribution data for training, and supervised models, which require additional information such as classification labels or sample OOD data. Unsupervised models are appealing as they make no assumptions about the form OOD data will take or the type of downstream task (e.g. classification, segmentation) that will be performed.

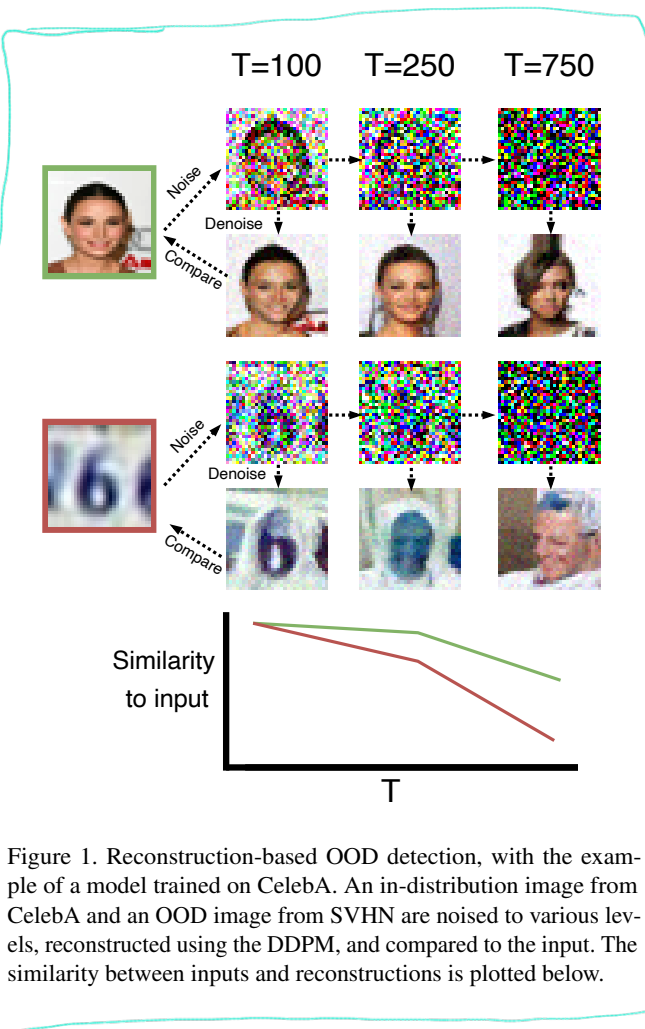


Figure 1. Reconstruction-based OOD detection, with the example of a model trained on CelebA. An in-distribution image from CelebA and an OOD image from SVHN are noised to various levels, reconstructed using the DDPM, and compared to the input. The similarity between inputs and reconstructions is plotted below.

The current dominant approach in unsupervised OOD detection is the use of the likelihood or other metrics from a generative model trained on the in-distribution data. However, it has been shown these models can exhibit egregious failures, such as a model trained on CIFAR10 assigning higher likelihoods to samples from the SVHN dataset

than samples from CIFAR10 itself [3, 9, 23]. A number of methods have been proposed to address these shortcomings [3, 22, 23, 29, 34]. These models have shown better performance in empirical benchmarks, but recent theoretical work suggests that all these methods will remain vulnerable against at least some OOD data [43].

**Reconstruction-based** methods offer an alternative approach to unsupervised OOD detection. They involve training a model to reconstruct in-distribution data and using the size of the reconstruction error to detect OOD inputs. However, compared to likelihood-based methods, reconstruction-based methods have received less attention in the literature [32, 42]. A likely reason is that these methods rely on an information bottleneck, such as a latent space that is smaller than the input, to effectively reconstruct in-distribution data but not OOD data. In practice, it is challenging to tune this bottleneck: too small and even the in-distribution data is poorly reconstructed, too large and even OOD data is successfully reconstructed. This need for tuning is undesirable and is likely a key reason these methods have typically been overlooked in favour of generative models for OOD detection.

Diffusion denoising probabilistic models (DDPM) [10, 35] present the information bottleneck issue in an interesting new light. These models are trained to incrementally remove noise from noised inputs. When an input is fully-noised, no information from the input itself is retained, and a DDPM will produce a new sample. However, when applying a DDPM to a partially-noised input, some information from the input is retained, and the denoising process is conditioned on that noisy input; that is, the model attempts to reconstruct the input. The amount of noise applied can be viewed as a variable information bottleneck. The bottleneck is not a property of the trained model itself, such as the size of the latent space in an autoencoder, but rather something we can control externally during model inference, meaning a single trained model can handle many different bottleneck levels. While this interpretation of DDPMs as autoencoders with an externally-controlled bottleneck has been previously discussed [5], to our knowledge, no work has attempted to use this property to perform reconstruction-based OOD detection with DDPMs.

In this work, we apply DDPMs to perform **reconstruction-based** OOD detection. We measure the quality of a model's reconstructions of an input noised to a range of different levels and propose to use this set of reconstruction error metrics to determine whether an image is OOD. Code will be made available upon publication; please contact the authors for earlier code access.

## 2. Related Work

Methods for OOD detection can be broadly categorised into supervised, requiring some additional labels or OOD

data, or unsupervised, requiring only in-distribution data. This overview of related work focuses on unsupervised methods.

### 2.1. Generative based

A conceptually appealing approach to OOD detection involves fitting a generative model  $p(\mathbf{x}; \theta)$  to a data distribution  $\mathbf{x}$  and evaluating the likelihood of unseen samples under this model. The assumption is that OOD samples will be assigned a lower likelihood than in-distribution samples and can be identified using a simple threshold on this value [2]. It has since been demonstrated that OOD samples will not necessarily be assigned lower likelihoods than in-distribution data; for example, different families of generative models trained on CIFAR10 all assign higher likelihoods to images from SVHN [3, 9, 23].

Subsequent studies have sought to address this shortcoming in generative models. One approach suggests the failures may be due to likelihood estimation errors and proposes using the Watanabe-Akaike Information Criterion (WAIC) across the likelihood estimates of an ensemble of models to identify OOD samples [3]. Other studies postulate that likelihood estimates are affected by population-level background statistics and propose using a likelihood ratio to remove this effect, either obtaining the denominator from an additional trained model [29] or a measure of image complexity [34]. A closely related approach notes that these methods do not perform well for Variational Autoencoders (VAE) and seeks to develop a specific likelihood ratio for this class of model [41]. Another related approach notes that lower-level model features dominate the likelihood and proposes the use of a hierarchical VAE and the requirement that a sample is in-distribution across all levels of the hierarchy [8]. Another strand of work proposes flagging samples as OOD if their likelihoods do not lie within the typical set of a model - put simply, a sample may be considered OOD not only if its likelihood is lower than that of in-distribution data, but also if it is higher [24]. Follow-up work proposes assessing the typicality of multiple summary statistics from the model, not just the likelihood, to assess whether samples are OOD [22].

### 2.2. Reconstruction based

Reconstruction-based methods represent another paradigm for OOD detection. They involve training a model  $R$  to reconstruct an input,  $\hat{\mathbf{x}} = R(\mathbf{x})$ . The intuition is that if  $R$  contains an information bottleneck, such as a latent space of lower dimension than the input size, it will only be capable of faithfully reconstructing inputs from the distribution it was trained on, and will poorly reconstruct OOD inputs. This can be measured using some similarity measure between the input and reconstruction,  $S(\mathbf{x}, \hat{\mathbf{x}})$ , and flagging inputs that show low similarity.

Several works have highlighted practical issues in the

use of reconstruction-based methods [4, 20, 27, 46]. If the information bottleneck is too small, a model cannot faithfully reconstruct even in-distribution samples. If the bottleneck is too large, the model is able to learn the identity function, allowing OOD samples to be reconstructed with low error. The result is that it is often necessary to perform a dataset-specific tuning process to produce a model that performs well, limiting the utility of such models in practice.

Some work has sought to address these issues. It has been suggested to use the Mahalanobis distance [21] in an autoencoder’s feature space as an OOD metric [4]. Other work has sought to reduce the ability of an autoencoder to reconstruct OOD samples by introducing a memory module and forcing the decoder to decode directly from the memory, encouraging any reconstructions to look more similar to in-distribution data [7]. However, neither of these works address the fundamental issue of bottleneck selection and require some tuning of the information bottleneck and/or memory size. We seek to address this shortcoming in our work with DDPM-based OOD detection.

### 3. Method

To enable reconstruction-based OOD detection that is not dependent on a fixed information bottleneck, we propose making use of a trained DDPM [10] to reconstruct images. During training, samples  $\mathbf{x}_0$  are degraded according to a fixed process with Gaussian noise according to a timestep  $t$  and a noise variance schedule  $\beta_t$  to produce noised samples  $\mathbf{x}_t$ , such that

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (1)$$

where  $0 \leq t \leq T$  and we define  $\alpha_t := 1 - \beta_t$  and  $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$ . The schedule  $\beta_t$  is designed to increase with  $t$  and have the property that the fully noised  $\mathbf{x}_T$  is close to an isotropic Gaussian,  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ; i.e.  $\mathbf{x}_T$  contains no information about  $\mathbf{x}_0$ . We train a single network to iteratively reverse the diffusion process by estimating the parameters of the denoising step given  $\mathbf{x}_t$  and  $t$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (2)$$

Given this trained model and a test input  $\mathbf{x}_0$ , we can sample a set of  $\mathbf{x}_t$  for a range of values of  $t$  and estimate their reconstructions,  $\hat{\mathbf{x}}_{0,t} = p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ . Measuring the similarity between each reconstruction and input  $S(\hat{\mathbf{x}}_{0,t}, \mathbf{x}_t)$  provides a range of similarity scores that can be used to decide whether  $\mathbf{x}_0$  is in-distribution.

The advantage of such a method over other reconstruction-based methods is that the information bottleneck is no longer a property of the network itself, such as the latent-space dimension in an autoencoder, but an externally chosen factor, the amount of noise applied to the input. This allows for

reconstructions from a wide number of information bottlenecks, obviating the need for the pre-selection of the appropriate bottleneck for a given dataset through the choice of model architecture.

#### 3.1. The diffusion model

We use the DDPM model from [30], which is a time-conditioned UNet [31]. While it is possible to optimise the variational bound on the negative log-likelihood, it has been found that a simplified training objective works well in practice, and we make use of it in this work. In this simplified scheme, the variance is fixed to time-dependent constants,  $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t \mathbf{I}$  and the network directly predicts the added noise  $\epsilon$  at step  $t$  [10]:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t)\|^2] \quad (3)$$

#### 3.2. Multiple reconstructions

Our method involves reconstructing an input  $\mathbf{x}_t$  for multiple values of  $t$ . In the DDPM sampling scheme,  $t$  steps are required to obtain the reconstruction  $\hat{\mathbf{x}}_{0,t}$ , with each step requiring an evaluation of the model. For a typical value  $T = 1000$  with reconstructions from 100 starting points equally spaced along the T-chain,  $T_{\text{start}} = [10, 20, 30, \dots, 990, 1000]$  we would need 50500 model evaluations to obtain all the reconstructions; equivalent to the computation required to obtain 50.5 samples from the model.

To expedite the process of obtaining multiple reconstructions, we make use of recent advances in fast sampling from diffusion models. In particular, we employ the PLMS sampler [18], which has been shown to substantially reduce the number of sampling steps whilst maintaining or even improving sample quality. If we select  $T = 1000$  but choose to use just 100 sampling steps, we are still able to perform 100 reconstructions. However, these reconstructions can be done with just 5050 model evaluations, equivalent to the compute required to obtain 5.05 samples from the model with a DDPM sampler, a  $10\times$  speed-up.

#### 3.3. Evaluating similarity

It is common to use the mean-squared error (MSE) between the input and reconstruction to evaluate similarity. In this work, we also choose to use the LPIPS metric, which uses the distance between the deep features of a network (in this case, Alexnet [15]) from two inputs as a measure of their perceptual similarity. LPIPS has been shown to correlate well with human evaluations of image similarity [45]. Using both MSE and LPIPS gives a total of  $2N$  similarity measurements per input for the  $N$  reconstructions performed. We convert each measurement into a Z-score using the measurements from a validation set for each reconstruction and

metric (MSE or LPIPS) separately. We average these  $2N$  Z-scores to produce an OOD score for each input.

### 3.4. Explainability maps

Whilst the primary aim of OOD methods is to accurately detect OOD inputs, a desirable secondary feature is the ability to highlight the parts of an image that are OOD. Reconstruction-based methods have the advantage of assessing similarity in image space, offering the possibility of flagging regions of the image that are OOD. We produce explainability maps by taking the squared-error between each reconstruction and input. We convert these to Z-scores using reconstructions on the validation set, and average the Z-score maps from all  $N$  reconstruction to produce a single spatial map per input image.

## 4. Experiments and Results

### 4.1. Experimental details

**Datasets.** We evaluate our method on a number of common computer vision benchmarks for OOD detection. We consider four in-distribution datasets: FashionMNIST [39], CIFAR10 [16], CelebA [19], and SVHN [25]. For the grayscale FashionMNIST, we use MNIST [17] as an OOD dataset. For the other colour datasets, we use all other colour datasets as OOD datasets. CelebA images were resized to 32x32 to match the dimension of the other colour datasets. We also use vertically- and horizontally-flipped versions of each in-distribution dataset as further OOD datasets, giving a total of 15 pairs of in vs out-of-distribution datasets.

**Baselines.** We benchmark against both generative- and reconstruction-based approaches. We used the Glow architecture [13] as the backbone for all generative approaches, following [22], and explored a number of methods in the literature to perform OOD using the trained model:

1. A threshold on the likelihood  $p(\mathbf{x}; \theta)$  as a simple baseline [2].
2. The Watanabe-Akaike Information Criterion (WAIC), obtained by training an ensemble of 5 models and calculating the WAIC score as  $\mathbb{E}_\theta[\log p(\mathbf{x}|\theta_n)] - \text{Var}_\theta[\log p(\mathbf{x}|\theta_n)]$  [3].
3. The single-sample typicality test where the typicality score for a sample  $\mathbf{x}$  is given by  $|\mathbb{H}[p(\mathbf{x}; \theta)] - \log p(\mathbf{x}; \theta)|$ , where  $\mathbb{H}[p(\mathbf{x}; \theta)]$  is calculated as an average over the training set [23].
4. Density-of-States Estimation (DoSE), to our knowledge the current state-of-the-art in unsupervised OOD detection. DoSE uses a density estimator to evaluate several features obtained from a model - for Glow models, the features are the model likelihood and its two constituent parts, the log-probability of the latent variable  $Z$ , and

the log of the determinant between the input and  $Z$ . We used principal component analysis (PCA) to learn a whitening transform and trained a one-class support vector machine (SVM) on the transformed features from the validation set and used it to score new samples [22]. As detailed in their paper, both the PCA and SVM used the default implementations in scikit-learn [26].

For the reconstruction-based approaches, we used:

1. A threshold on the MSE from an autoencoder-based (AE) reconstruction  $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$ .
2. A threshold on the Mahalanobis score, given by  $\alpha D_M(E(\mathbf{x})) + \beta \|\mathbf{x} - \hat{\mathbf{x}}\|_2$  where  $D_M(\mathbf{x})$  is the Mahalanobis distance from the latent space of  $\mathbf{x}$  to the latent space of the training set and  $\alpha$  and  $\beta$  are constants set to the reciprocal of the standard deviation of  $D_M(E(\mathbf{x}))$  and  $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$  respectively, as evaluated on the validation set [4].
3. The MSE from reconstructions using a memory-augmented autoencoder (MemAE), which augments the standard AE with a memory bank and seeks to decode samples from atomic elements in this bank [7].

**Implementation details.** For our method, we used the DDPM model as described in [30]<sup>1</sup>. We used a 3-layer UNet with [256, 512, 784] channels, with two residual blocks per layer and a single-headed attention block after each residual block in layers 2,3 of the downsampling branch and layer 1 of the upsampling branch. The timestep was sinusoidally embedded and passed through a two-layer MLP with a Swish activation function [28] to create a 1024-dim embedding. We used  $T = 1000$  during training and a linear noise schedule with  $\beta_t$  varying between 0.0015 and 0.0195. All models were trained for 300 epochs using the Adam optimiser [12] with a learning rate of  $2.5e^{-5}$ . At test time, we used the PLMS sampler with 100 timesteps and reconstructed from each of these 100 steps as starting points to produce 100 reconstructions per input.

We implemented and trained all baselines to enable comparisons across the full set of dataset pairings considered, using the author’s implementations when available. We trained Glow models<sup>2</sup> using the architectural details outlined in [22, 24]: we used 3 blocks (except for the FashionMNIST model, which used 2), each with 8 layers. Each layer used an activation norm, an inverted  $1 \times 1$  convolution and affine coupling with 400 hidden channels per layer. Each model was trained for 100 epochs with a batch size of 64, using the Adamx optimiser with a learning rate of  $5e^{-4}$ , weight decay of  $5e^{-5}$  and a 10 epoch warmup.

<sup>1</sup><https://github.com/CompVis/latent-diffusion/>

<sup>2</sup><https://github.com/y0ast/Glow-PyTorch>



	FashionMNIST				CIFAR10				CelebA				SVHN				Rank
	MNIST	VFlip	HFlip	SVHN	CelebA	VFlip	HFlip	CIFAR10	SVHN	VFlip	HFlip	CIFAR10	CelebA	VFlip	HFlip		
<b>Generative-based</b>																	
Likelihood [2]	8.5	55.5	51.1	6.1	52.3	50.5	50.0	67.4	5.9	57.9	50.1	99.0	<b>99.9</b>	50.2	50.3	4.6	
WAIC [3]	8.8	55.4	51.1	6.0	52.5	50.4	50.0	67.0	5.69	57.8	50.1	99.0	<b>99.9</b>	50.2	50.3	4.9	
Typicality [23]	81.1	51.2	49.6	88.6	40.3	50.0	50.0	66.5	88.8	51.2	50.0	97.1	99.8	50.0	49.9	6.0	
Density-of-States [22]	<b>98.1</b>	67.1	55.3	96.4	54.6	51.0	50.1	84.9	99.4	69.6	49.9	98.9	<b>99.9</b>	50.3	50.9	<b>3.0</b>	
<b>Reconstruction-based</b>																	
AutoEncoder	75.0	59.0	50.4	3.2	<b>75.3</b>	50.1	49.9	42.1	2.7	53.2	49.9	<b>99.4</b>	<b>99.9</b>	49.9	49.9	5.7	
AutoEncoder Mahlabonis [4]	94.9	79.5	63.0	4.46	71.8	50.9	50.0	64.2	9.6	71.6	50.1	99.3	99.8	50.6	50.6	<b>3.3</b>	
MemAE [7]	56.9	59.0	48.7	4.21	69.4	50.3	49.9	51.5	5.8	56.4	49.9	98.6	99.5	49.8	49.7	6.6	
DDPM (ours)	97.4	<b>88.6</b>	<b>65.1</b>	<b>97.9</b>	68.5	<b>63.2</b>	<b>50.5</b>	<b>99.0</b>	<b>100</b>	<b>93.3</b>	<b>50.3</b>	99.0	99.6	<b>58.2</b>	<b>61.6</b>	<b>1.9</b>	

Table 1. AUC score for each comparison. Bold text indicates the highest value per column. The overall rank is calculated as the average across the ranks for each of the 15 comparisons (lower is better).

We trained<sup>3</sup> three-layer AEs with [128, 128, 256] features in the encoder and [128, 128, 3] in the decoder for colour datasets, and [32, 16, 8] in the encoder and [16, 32, 1] in the decoder for grayscale as in [7]. Each layer is followed by a batch normalisation [11] and a leaky ReLU activation, with upsampling implemented using transposed convolutions. The MemAE contained an additional memory module with sparse shrinkage. Following [7], we used a memory size of 100 for the grayscale datasets and 500 for colour datasets, with a shrinkage threshold of  $2.5e^{-3}$ . Models were trained for 100 epochs with Adam with a learning rate of  $1e^{-4}$  and a batch size of 10.

## 4.2. Comparison to other methods

A summary of the quantitative results for all baselines is presented in Table 1, reported as AUC scores. The DDPM has the highest average rank across all experiments, with the state-of-the-art DoSE ranked second-highest. In a direct comparison, the DDPM outperforms DoSE on 13/15 dataset pairings. The increases in performance afforded by the DDPM are sometimes substantial, most notably when the OOD dataset is a vertically- or horizontally-flipped version of the in-distribution dataset. For example, on FashionMNIST vs VFlip, DDPM improves performance over DoSE from an AUC of 67.1 to 88.6 and on HFlip from 55.3 to 65.1. The DDPM is the only method able to perform substantially better than chance on certain pairings: for example, no other method scores higher than 50.9 on either SVHN vs VFlip or HFlip, whilst DDPM scores 58.2 and 61.6, respectively.

Fig. 2 shows some reconstructions from the model trained on the SVHN dataset. Reconstructions from all four models are included in Supplementary Fig. 7. Reconstructions of the in-distribution SVHN input still retain similarity to the

input up until noising to  $t = 500 - 600$ , whilst the OOD reconstructions start to look dissimilar to their inputs after noising to  $t = 100$  and bear almost no resemblance by  $t = 400$ . The plot also shows that for  $t \gtrsim 700$  the noised images retain very little information from the input and the model outputs begin to resemble unconditioned samples more than they do reconstructions. This suggests that reconstructions from higher  $t$  contribute little to the OOD signal and can potentially be discarded. We explore this further in Sec. 4.3.

As widely reported [3, 23, 29], using the likelihood alone performs very poorly on a number of dataset pairings, such as FashionMNIST vs MNIST and CIFAR10 vs SVHN. Typicality substantially improves performance on the three dataset pairings that simple likelihood performs the worst on, but offers little improvements on other dataset pairings. In agreement with [22] we found that WAIC provides little advantage over using just the likelihood, and sometimes degrades performance.

All three autoencoder-based reconstruction methods exhibit inconsistent performance. They perform well on SVHN vs CIFAR10/CelebA but completely fail on other pairings, such as CIFAR10 vs SVHN. Inspection of the results reveals this class of method tends to perform better when the in-distribution datasets have simpler textures than the OOD datasets, as this causes higher OOD reconstruction error - this explains the good performance on SVHN vs CIFAR10/CelebA. Conversely, when in-distribution datasets are more complicated, the models can reconstruct simpler OOD datasets with very low error and perform poorly at OOD detection, as in CIFAR10 vs SVHN.

To explore the importance of bottleneck size for autoencoder models, we experimented with decreasing the size of the latent space with 4-layer architectures, and both increasing and decreasing the memory size for MemAE models. We found that different sizes could improve performance on

<sup>3</sup><https://github.com/donggong1/memae-anomaly-detection>

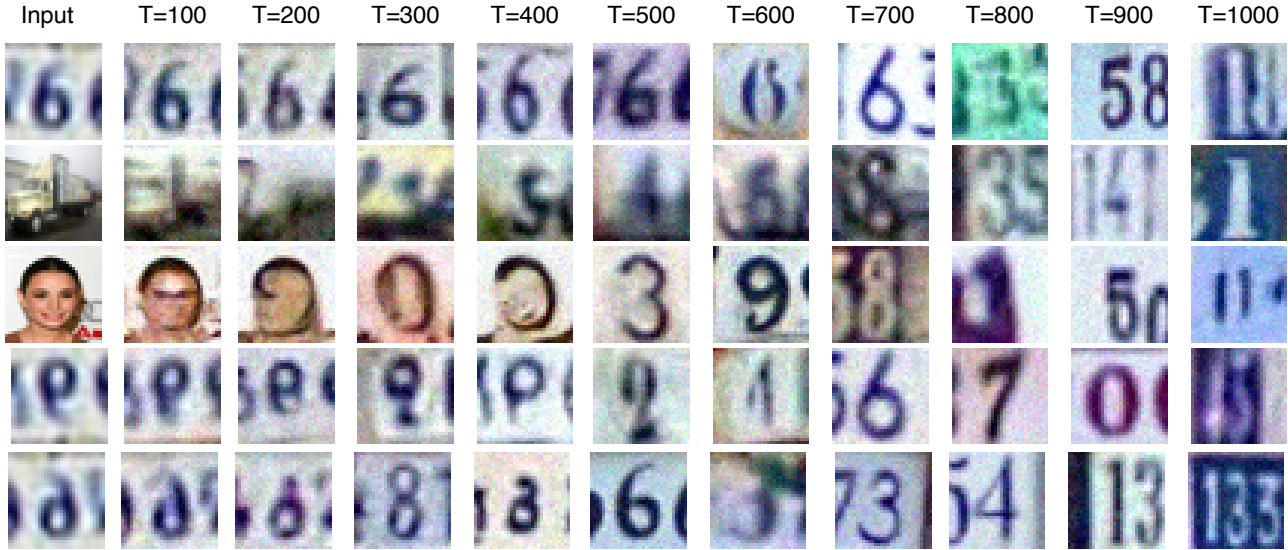


Figure 2. Example reconstructions from a model trained on SVHN for ten different  $t$ -values spaced equally across the chain. Plot shows an in-distribution input (top row) and OOD inputs from CIFAR10, CelebA, SVHN VFlip, and SVHN HFlip (rows 2-5).

some datasets, but at the expense of performance on others. For example, a 4-layer MemAE with a memory size of 2000 achieved 91.5 on FashionMNIST vs MNIST, substantially higher than the 3-layer model with recommended memory size, but this architecture’s performance on SVHN vs CIFAR10 dropped to 89.3, making it the poorest-performing of any model (not just MemAE models) on this pairing. These results highlight the need for undesirable dataset-specific tuning for standard reconstruction-based OOD detection. Full results are included in the Supplementary Table 3.

#### 4.3. Performance and number of reconstructions

We evaluated the performance of the DDPM model as the number of reconstructions performed was reduced. Results are shown in Fig. 3 with a full breakdown by dataset pairing included in Supplementary Table 4. Reconstruction starting points were uniformly sub-sampled across the range of possible starting points; e.g. to reduce from 100 to 25 reconstructions we selected every fourth starting point. The results show that we can reduce the number of reconstructions from 100 to 25 with a mean drop in AUC across experiments of 0.58 (with AUC scores reported in the range of 0-100). Performing 25 reconstructions requires 1225 model evaluations, only slightly more than required to obtain a single sample from the model using the DDPM sampler over 1000 steps. Dropping even further to just 13 reconstructions gives a mean drop in AUC of 1.49, still outperforming the state-of-the-art DoSE method on 13/15 dataset pairings, but requiring just 637 evaluations, i.e. slightly over half those required to obtain a single sample from the model. Reducing the number of reconstructions below this point begins to

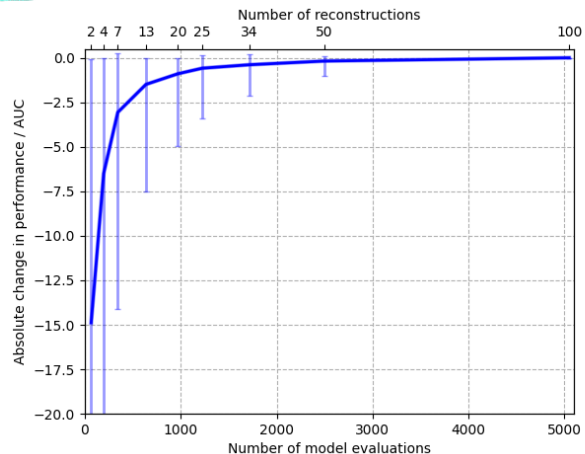


Figure 3. Results show the mean, minimum and maximum drop in absolute AUC as the number of reconstructions used to perform OOD detection is reduced, compared to results obtained using 100 reconstructions. The number of model evaluations is plotted on the lower x-axis, as this quantity directly correlates with total inference time. Full results are tabulated in the supplement.

affect performance substantially.

We further explored reducing the maximum value of  $t$  that reconstructions were performed from,  $\max_T$ , motivated by the observation in Fig. 2 that reconstructions from higher  $t$ -values do not seem heavily conditioned on the input and might not add to the OOD signal. Fig. 4 shows the mean drop in dataset performance, compared to the original model reported in Table 1, as  $\max_T$  was reduced from 1000, and

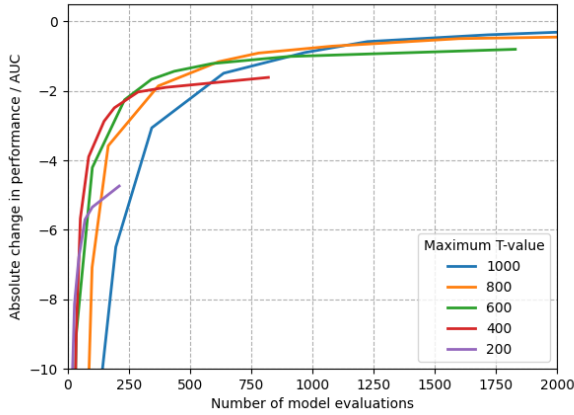


Figure 4. Results show the mean drop in absolute AUC, relative to the model with  $\max_T = 1000$  and 100 reconstructions, as both  $\max_T$  and the number of reconstructions performed for a given  $\max_T$  are reduced.

as the number of reconstructions was reduced for a given value of  $\max_T$ . The results show that we can do better on the model-evaluations vs performance curve by reducing  $\max_T$ , meaning for a limited compute budget it is better to focus on reconstructions from lower  $t$ -values. However, this result is likely dependent on the size of the input. There is a relationship between the amount of noise applied to an image and the size of features that are detectable in that image [5], and we suspect that as input size increases, the maximum value of  $t$  that provides useful OOD signal will also increase. Reducing  $\max_T$  did not lead to mean performance better than the model with  $\max_T = 1000$  and 100 reconstructions, suggesting our approach is robust to the inclusion of non-informative signal and that there is no need for dataset-specific tuning of the bottlenecks. Detailed results are tabulated in Supplementary Table 4.

#### 4.4. Variants of the proposed method

We investigated the effects of changing the method used to classify OOD samples. We first investigated using a one-class SVM or a Gaussian Mixture Model (GMM), trained on the validation, set to score samples. Results in Table 2 show that, while all methods perform well, the GMM and SVM are consistently outperformed by the simple Z-score averaging method. Fig. 5 shows the Z-scores for a dataset pairing where Z-score averaging substantially outperforms an SVM or GMM (FashionMNIST vs FashionMNIST HFlip) and a pairing where the difference is small (SVHN vs CelebA). These plots suggest that Z-score averaging has an advantage when the signals from in-distribution and OOD inputs are noisy and overlap substantially, suggesting that the GMM/SVM overfit to these noisy signals, impairing classi-

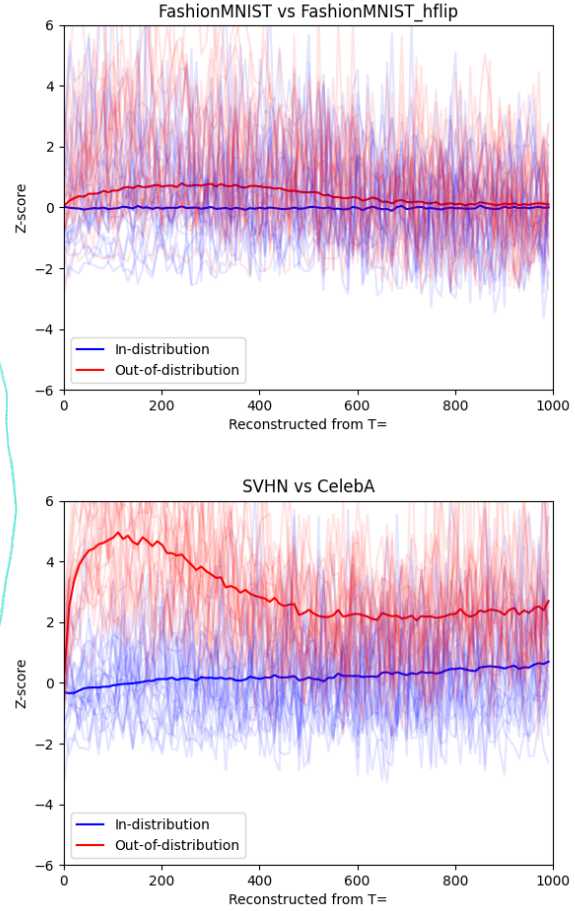


Figure 5. Plots of Z-scores of similarity for all 100 reconstructions of an image. Thin lines represent plots from individual inputs, thick lines show the means of these values. For FashionMNIST vs FashionMNIST HFlip (top figure), Z-score averaging performs much better than an SVM or GMM, whilst for SVHN vs CelebA (bottom figure), the difference in performance is small.

cation performance. Table 2 also shows performance when using MSE or LPIPS alone as similarity metrics, compared to MSE+LPIPS combined. Results indicate these two metrics are complementary, and their combination tends to improve performance.

#### 4.5. Explainability maps

A desirable secondary characteristic of an OOD model is the ability to highlight the regions of an image that are OOD. To explore the ability of our model to produce explainability maps, we took in-distribution images and synthesised anomalies by randomly sampling rectangular blocks, and adding them to a randomly selected colour channel in colour images, or deleting the block in grayscale images. We produced maps following the procedure in Sec. 3.4, with results in Fig. 6.

	FashionMNIST			CIFAR10				CelebA				SVHN			
	MNIST	VFlip	HFlip	SVHN	CelebA	VFlip	HFlip	CIFAR10	SVHN	VFlip	HFlip	CIFAR10	CelebA	VFlip	HFlip
Metric: MSE+LPIPS															
Classification method:															
GMM	95.1	82.3	60.8	96.8	55.8	53.1	49.8	98.3	100	88.9	50.1	97.6	98.6	52.3	55.1
SVM	95.6	82.5	60.7	96.4	56.8	52.8	49.9	97.5	100	87.6	50.0	97.8	99.2	52.6	55.1
Z-score average	97.4	88.6	65.1	97.9	68.5	63.2	50.5	99.0	100	93.3	50.3	99.0	99.6	58.2	61.6
Classification method: Z-score average															
Metric:															
MSE	92.4	83.6	63.8	40.2	70.3	57.2	50.5	86.3	77.5	86.5	50.2	97.3	99.7	58.8	60.8
LPIPS	95.4	81.5	61.6	98.5	50.5	58.6	50.3	98.3	100	87.6	50.2	92.3	87.5	53.4	56.0
MSE+LPIPS	97.4	88.6	65.1	97.9	68.5	63.2	50.5	99.0	100	93.3	50.3	99.0	99.6	58.2	61.6

Table 2. AUC score for variants of the DDPM method. In the top set of results, the similarity metrics used are fixed to MSE+LPIPS, and we try three classification methods: a GMM, SVM, and a Z-score average. In the bottom set, the classification method is fixed to Z-score averaging, and we use three different similarity metrics: MSE, LPIPS, and MSE+LPIPS.

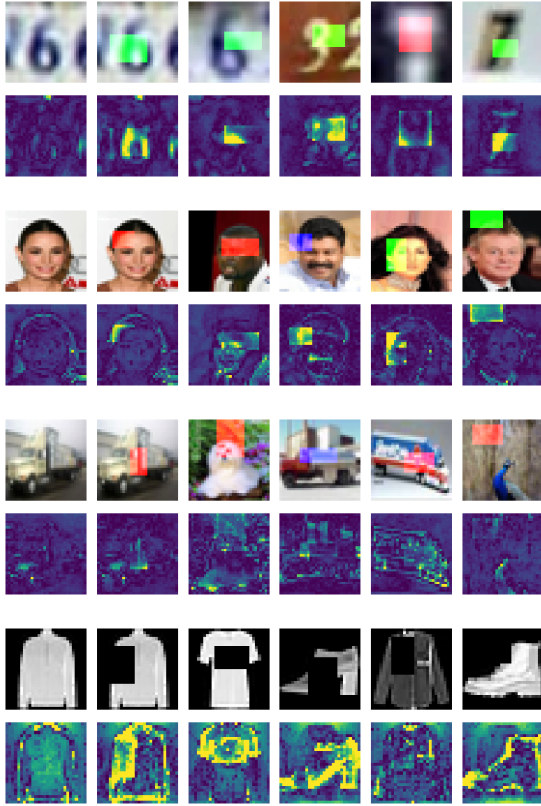


Figure 6. Top rows show inputs, and bottom rows show corresponding explainability maps plotted as Z-scores in the range [0-2].

The results suggest the ability to localise anomalies is dependent on the complexity of the in-distribution dataset. The maps successfully highlight the abnormalities for the SVHN

and CelebA datasets, but perform poorly on CIFAR10 and FashionMNIST. These latter two datasets have greater variability in the placement and/or appearance of the foreground object, which introduces more noise into the reconstructions and consequently suppresses the anomaly localisation.

#### 4.6. Limitations

A key limitation of the method is that it is more computationally expensive than competing methods. This might limit the contexts for which the method can be used, even with the potential efficiency enhancements discussed in Sec. 4.3. For example, the method may be suitable in medical imaging applications where it may be acceptable to flag an input as OOD within minutes, but less suitable in self-driving cars where predictions need to be made in real time. However, there is considerable scope for improving the efficiency of these models. The field of DDPMs is currently very active with substantial interest in fast sampling, and our method can directly benefit from advances in sampling speed [1, 6, 14, 33, 36–38, 40, 44]. Finally, while the steps along a single reconstruction chain cannot be parallelised, reconstructions from different starting points can be. This could reduce the reconstruction time down to the amount required to reconstruct the longest chain.

#### 5. Conclusion

In this work, we explored how DDPMs can be used to perform unsupervised OOD detection. We propose performing reconstructions of a number of inputs noised to different extents, addressing a drawback of standard reconstruction techniques that require the choice of a single, fixed bottleneck. To the best of our knowledge, this is the first application of DDPMs to OOD detection. We tested our method with a range of OOD benchmarks. These experi-



ments show that our DDPM-based method outperforms not only reconstruction-based methods but also state-of-the-art generative approaches.

## References

- [1] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2021. 8
- [2] Christopher M Bishop. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994. 2, 4, 5
- [3] Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018. 2, 4, 5
- [4] 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. 3, 4, 5
- [5] Sander Dieleman. Diffusion models are autoencoders, 2022. 2, 7
- [6] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Genie: Higher-order denoising diffusion solvers. *arXiv preprint arXiv:2210.05475*, 2022. 8
- [7] 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*, pages 1705–1714, 2019. 3, 4, 5, 12
- [8] Jakob D Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical vaes know what they don’t know. In *International Conference on Machine Learning*, pages 4117–4128. PMLR, 2021. 2
- [9] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2018. 2
- [10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3
- [11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 5
- [12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4
- [13] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. 4
- [14] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021. 8
- [15] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014. 3
- [16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4
- [17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4
- [18] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021. 3
- [19] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 4
- [20] Olga Lyudchik. Outlier detection using autoencoders. Technical report, 2016. 3
- [21] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936. 3
- [22] 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*, pages 3232–3240. PMLR, 2021. 2, 4, 5
- [23] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don’t know? In *International Conference on Learning Representations*, 2018. 2, 4, 5
- [24] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*, 2019. 2, 4
- [25] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 4
- [26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4
- [27] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014. 3
- [28] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 4
- [29] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019. 2, 5
- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 3, 4

- [31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3
- [32] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*, 2021. 2
- [33] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2021. 8
- [34] Joan Serrà, David Álvarez, Vicens Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2019. 2
- [35] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2
- [36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 8
- [37] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021. 8
- [38] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2021. 8
- [39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 4
- [40] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. In *International Conference on Learning Representations*, 2021. 8
- [41] 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. 2
- [42] Jingkan Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021. 2
- [43] Lily Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *International Conference on Machine Learning*, pages 12427–12436. PMLR, 2021. 2
- [44] Qingsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022. 8
- [45] 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*, pages 586–595, 2018. 3
- [46] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018. 3

## A. Sample reconstructions from all models

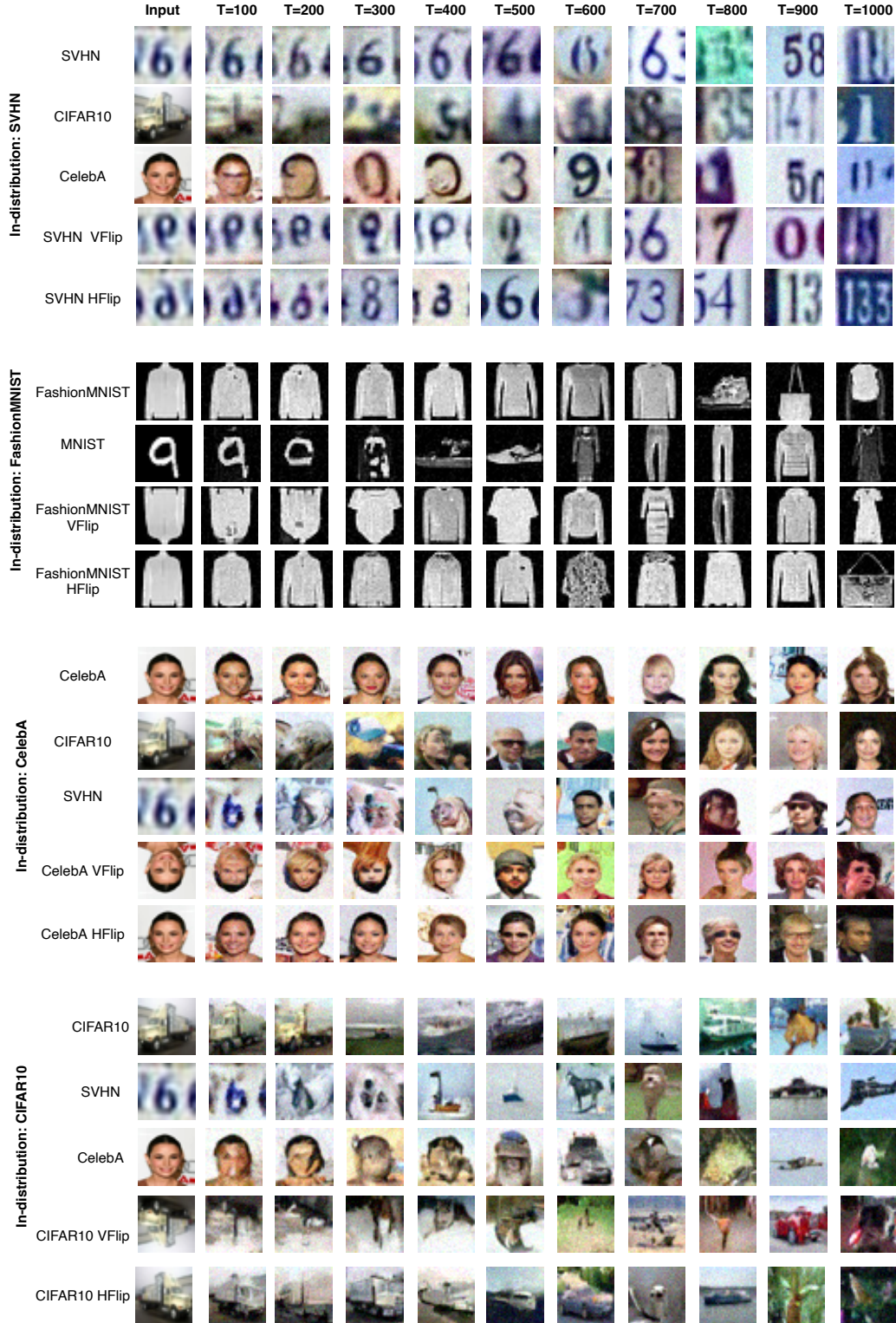


Figure 7. Example reconstructions from all models from ten different starting points spanning the full T-chain.

	FashionMNIST			CIFAR-10				CelebA				SVHN			
	MNIST	VFlip	HFlip	SVHN	CelebA	VFlip	HFlip	CIFAR10	SVHN	VFlip	HFlip	CIFAR10	CelebA	VFlip	HFlip
AutoEncoder	75.0	59.0	<b>50.4</b>	<b>3.2</b>	<b>75.3</b>	50.1	<b>49.9</b>	42.1	2.7	53.2	<b>49.9</b>	<b>99.4</b>	<b>99.9</b>	49.9	<b>49.9</b>
AutoEncoder (4layer)	<b>95.1</b>	<b>71.9</b>	49.5	2.7	66.4	<b>50.4</b>	<b>49.9</b>	<b>65.2</b>	<b>6.7</b>	<b>68.5</b>	<b>49.9</b>	98.8	99.5	<b>50.3</b>	<b>49.9</b>
AutoEncoder Mahlabonis	94.9	<b>79.5</b>	63.0	4.5	<b>71.8</b>	50.9	<b>50.0</b>	64.2	9.6	71.6	<b>50.1</b>	<b>99.3</b>	<b>99.8</b>	50.6	<b>50.6</b>
AutoEncoder Mahlabonis (4layer)	<b>98.3</b>	77.7	<b>64.1</b>	<b>6.8</b>	69.0	<b>51.4</b>	<b>50.0</b>	<b>73.3</b>	<b>19.5</b>	<b>78.7</b>	<b>50.1</b>	96.9	98.6	<b>51.1</b>	<b>50.6</b>
MemAE (mem size rec)	56.9	59.0	48.7	4.21	<b>69.4</b>	50.3	49.9	51.5	5.8	56.4	49.9	98.6	99.5	49.8	49.7
MemAE (mem size 2000)	88.4	58.2	49.1	26.8	61.2	49.3	50.0	67.2	36.0	63.1	<b>50.0</b>	95.2	98.5	50.1	<b>50.7</b>
MemAE (mem size 50)	28.6	53.0	49.9	7.2	67.1	<b>50.7</b>	49.8	63.5	33.3	53.5	49.8	<b>98.7</b>	<b>99.6</b>	50.1	50.0
MemAE (4layer, mem size rec)	77.3	61.0	51.1	7.1	66.1	50.6	<b>50.1</b>	75.1	25.4	73.3	49.9	97.1	98.6	50.6	50.2
MemAE (4layer, mem size 2000)	<b>91.5</b>	<b>71.4</b>	<b>52.5</b>	<b>33.1</b>	66.4	<b>50.7</b>	<b>50.1</b>	<b>78.6</b>	<b>55.8</b>	<b>80.3</b>	<b>50.0</b>	89.3	94.5	<b>52.5</b>	50.4
MemAE (4layer, mem size 50)	87.6	65.2	51.3	7.0	62.3	50.6	50.0	71.7	23.8	64.4	<b>50.0</b>	97.2	98.7	50.3	50.2

Table 3. AUC score for each comparison. Bold text indicates highest value per column, done separately for each model class. Recommended memory size means 100 for FashionMNIST and 500 for other datasets, according to the settings in [7].

## B. Performance of AEs vs model size

Full results are shown in Table 3. The results highlight that tuning the information bottleneck can improve performance on a given dataset, but at the expense of performance on other datasets.

## C. Performance vs number of reconstructions

Results in Table 4. They demonstrate the number of reconstructions/model evaluations can be reduced substantially with limited effect on OOD performance.



Recons	Model evals	FashionMNIST			CIFAR10				CelebA				SVHN			
		MNIST	VFlip	HFlip	SVHN	CelebA	VFlip	HFlip	CIFAR10	SVHN	VFlip	HFlip	CIFAR10	CelebA	VFlip	HFlip
max <sub>t</sub> = 1000																
100	5050	97.4	88.6	65.1	97.9	68.5	63.2	50.5	99.0	100.0	93.3	50.3	99.0	99.6	58.2	61.6
50	2500	97.1	88.1	65.1	97.8	67.8	63.0	50.5	98.9	100.0	92.9	50.3	99.0	99.5	58.0	61.4
34	1717	97.0	87.3	64.5	97.7	67.3	62.9	50.7	98.8	100.0	92.7	50.5	98.8	99.4	57.8	60.9
25	1225	96.7	86.9	64.7	97.6	66.2	62.3	50.6	98.7	100.0	92.2	50.3	98.7	99.4	57.9	61.1
20	970	96.4	86.2	64.0	97.3	65.3	62.0	50.3	98.6	100.0	91.8	50.3	98.6	99.3	57.7	60.8
13	637	95.4	84.6	63.8	97.2	63.9	61.0	50.4	98.3	100.0	90.7	50.2	98.2	98.9	57.0	60.2
7	343	91.7	80.6	62.3	96.6	61.1	59.4	50.7	97.3	99.9	87.4	50.4	96.6	97.4	55.9	58.7
4	196	82.2	73.5	60.0	95.2	56.4	56.4	50.4	94.7	99.8	80.9	50.1	92.0	93.0	54.0	55.9
2	66	39.0	60.7	54.8	91.8	46.5	52.6	50.0	87.8	98.8	69.7	50.2	83.3	80.0	51.4	52.3
max <sub>t</sub> = 800																
80	3240	97.2	88.6	65.3	96.7	62.5	62.1	50.3	98.7	100.0	92.7	50.3	99.2	99.5	60.2	64.0
40	1600	96.9	88.1	65.3	96.6	62.0	62.0	50.4	98.6	100.0	92.4	50.3	99.0	99.4	59.9	63.7
27	1080	96.7	87.5	64.7	96.5	61.4	61.8	50.4	98.5	100.0	92.1	50.5	98.9	99.2	59.7	63.3
20	780	96.4	87.1	65.0	96.3	60.7	61.4	50.5	98.4	100.0	91.7	50.3	98.8	99.1	59.6	63.3
16	616	96.0	86.5	64.4	96.1	60.0	61.3	50.3	98.3	99.9	91.3	50.3	98.7	99.0	59.5	63.0
10	370	94.8	85.1	64.1	95.7	58.0	60.2	50.2	97.9	99.9	90.0	50.2	98.2	98.5	58.9	62.5
5	165	90.3	81.7	62.9	94.7	54.0	58.7	50.4	96.4	99.9	86.6	50.4	96.6	96.3	58.1	61.2
3	99	80.4	74.9	60.3	93.6	50.5	56.2	50.3	93.7	99.7	80.3	50.1	92.2	90.6	55.5	57.6
2	66	39.0	60.7	54.8	91.8	46.5	52.6	50.0	87.8	98.8	69.7	50.2	83.3	80.0	51.4	52.3
max <sub>t</sub> = 600																
60	1830	96.6	88.4	65.2	95.0	56.3	61.4	50.3	98.2	99.9	92.1	50.3	99.0	99.3	62.0	66.0
30	900	96.2	88.0	65.2	94.8	55.9	61.2	50.4	98.1	99.9	91.7	50.3	98.9	99.1	61.6	65.6
20	590	95.9	87.5	64.8	94.7	55.2	61.1	50.4	97.9	99.9	91.3	50.5	98.7	98.9	61.5	65.4
15	435	95.4	87.0	64.8	94.6	54.9	60.8	50.4	97.8	99.9	90.9	50.4	98.6	98.8	61.2	65.0
12	342	94.9	86.5	64.4	94.4	54.4	60.9	50.2	97.6	99.9	90.3	50.4	98.4	98.7	61.2	64.8
8	232	93.7	85.2	64.0	94.3	53.8	60.0	50.3	97.3	99.9	89.4	50.3	98.0	98.1	60.1	63.6
4	100	87.9	81.7	62.7	93.2	50.1	58.6	50.5	95.6	99.8	85.4	50.4	96.2	95.6	59.0	62.1
2	34	72.4	75.0	59.9	90.5	43.1	56.0	50.3	91.0	99.2	76.6	50.1	90.3	87.1	56.7	58.6
max <sub>t</sub> = 400																
40	820	93.2	87.6	64.3	92.9	50.5	61.4	50.3	97.5	99.9	90.7	50.4	98.8	99.2	63.6	67.5
20	400	92.3	87.1	64.1	92.7	50.1	61.2	50.4	97.3	99.9	90.1	50.4	98.7	99.0	63.2	66.9
14	287	92.3	86.7	63.8	92.8	50.1	61.2	50.3	97.2	99.9	89.8	50.5	98.5	98.8	62.9	66.7
10	190	90.2	86.0	63.6	92.4	49.1	60.8	50.5	97.0	99.8	89.0	50.5	98.4	98.7	62.7	66.2
8	148	88.8	85.3	63.3	92.3	48.6	60.7	50.1	96.7	99.8	88.3	50.2	98.2	98.4	62.5	65.8
5	85	84.4	83.7	62.6	91.7	47.0	59.9	50.4	96.0	99.7	86.4	50.3	97.6	97.6	61.7	64.7
3	51	79.8	80.6	61.6	91.3	45.5	58.4	50.3	94.4	99.6	82.7	50.2	95.7	94.7	59.8	62.5
2	34	72.4	75.0	59.9	90.5	43.1	56.0	50.3	91.0	99.2	76.6	50.1	90.3	87.1	56.7	58.6
max <sub>t</sub> = 200																
20	210	69.7	82.8	61.7	90.8	45.6	60.5	50.3	96.2	99.5	87.4	50.3	98.4	99.1	62.7	65.7
10	100	66.5	81.9	61.4	90.5	45.0	60.2	50.3	95.8	99.5	86.4	50.3	98.1	98.9	62.1	64.9
7	70	64.8	81.3	61.1	90.5	44.7	60.1	50.4	95.6	99.5	85.6	50.3	97.8	98.5	61.7	64.6
5	45	58.0	79.7	60.5	90.1	43.4	59.2	50.3	95.0	99.3	84.1	50.5	97.5	98.1	61.1	63.5
4	34	53.6	78.5	60.3	90.0	42.6	59.0	50.0	94.5	99.2	82.8	50.1	97.0	97.5	60.3	62.6
3	27	53.0	77.7	59.7	89.7	42.0	58.3	50.2	93.8	99.2	81.1	50.4	96.3	96.2	60.0	62.4
2	18	46.3	74.6	58.4	89.0	40.2	57.0	50.1	91.6	99.0	76.5	50.3	93.8	91.9	58.5	60.7

Table 4. Variation in the performance of the DDPM OOD detection as the number of reconstructions used is changed. Values reported for each dataset pairing are AUCs. The first row ( $\max_t = 1000$ , 100 reconstructions) are the parameters for the results reported in Table 1.