

3 / 15 / 2024 (Fri)

# **Unsupervised Out-of-Distribution Detection using Diffusion and Consistency Models**

**Seokho Moon**

danny232@korea.ac.kr

School of Industrial and Management Engineering, Korea University



### 문석호 (Seokho Moon)

- 고려대학교 산업경영공학과 대학원 재학 중
- Data Mining & Quality Analytics Lab (김성범 교수님)
- 석박통합과정 (2019.09 ~ )

### 관심 연구 분야

- Out-of-distribution detection (anomaly detection)
- Generative models

### E-mail

- danny232@korea.ac.kr

# Introduction (Out-of-Distribution Detection)

- **Definitional Overlap in Out-of-Distribution Detection**

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

# Introduction (Out-of-Distribution Detection)

- **Definitional Overlap in Out-of-Distribution Detection**

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

● : In-distribution

In-distribution (normal)





# Introduction (Out-of-Distribution Detection)

- **Definitional Overlap in Out-of-Distribution Detection**

Anomaly detection, out-of-distribution detection, open-set recognition ...

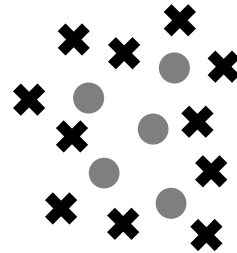
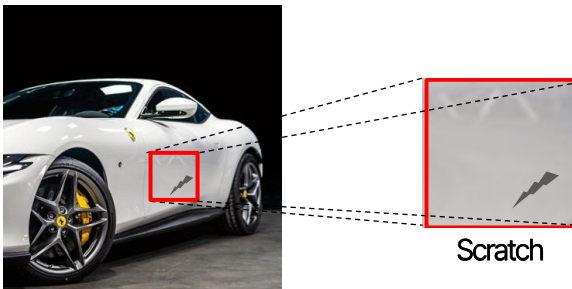
Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

In-distribution (normal)



Out-of-distribution (anomaly)



● : In-distribution

✕ : Out-of-distribution

# Introduction (Out-of-Distribution Detection)

- **Definitional Overlap in Out-of-Distribution Detection**

Anomaly detection, out-of-distribution detection, open-set recognition ...

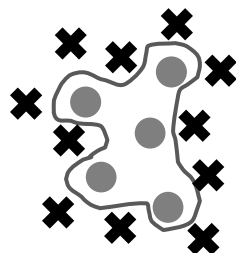
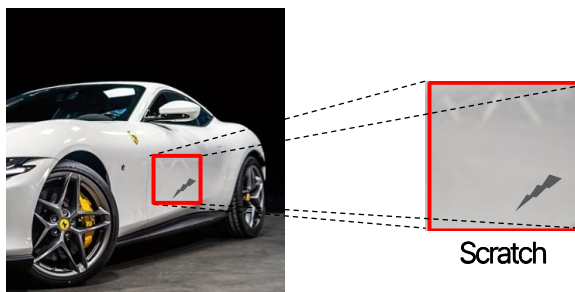
Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

In-distribution (normal)



Out-of-distribution (anomaly)



● : In-distribution  
✕ : Out-of-distribution

## Anomaly detection

→ in-distribution 과 out-of-distribution 의 미세한 차이 탐지  
(covariate shift)

# Introduction (Out-of-Distribution Detection)

- **Definitional Overlap in Out-of-Distribution Detection**

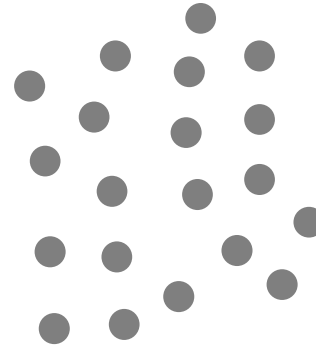
Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

● : In-distribution

In-distribution (vehicles)



# Introduction (Out-of-Distribution Detection)

- **Definitional Overlap in Out-of-Distribution Detection**

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

In-distribution (vehicles)



Out-of-distribution

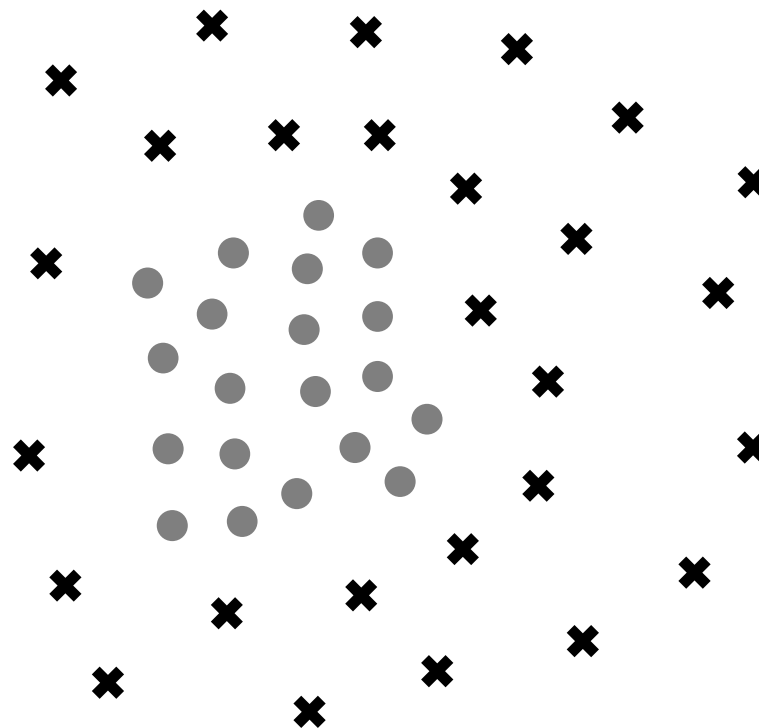
Near-OOD



Far-OOD



● : In-distribution  
✕ : Out-of-distribution



# Introduction (Out-of-Distribution Detection)

- Definitional Overlap in Out-of-Distribution Detection

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

In-distribution (vehicles)



Out-of-distribution

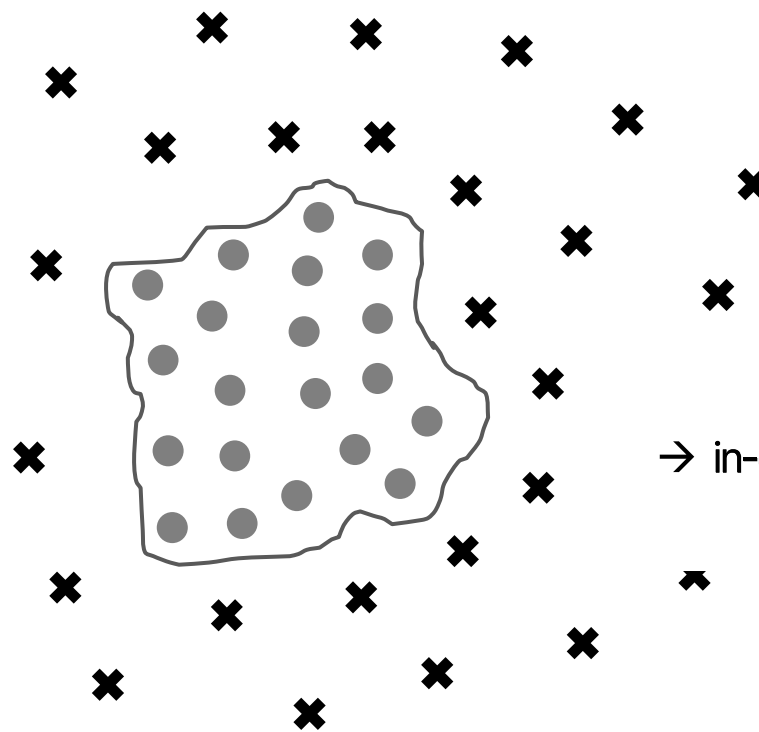
Near-OOD



Far-OOD



● : In-distribution  
✕ : Out-of-distribution



## Out-of-distribution detection

→ in-distribution 과 out-of-distribution 의 분포 차이 탐지  
(semantic shift)



# Introduction (Out-of-Distribution Detection)

- **Definitional Overlap in Out-of-Distribution Detection**

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

In-distribution (vehicles)



Out-of-distribution

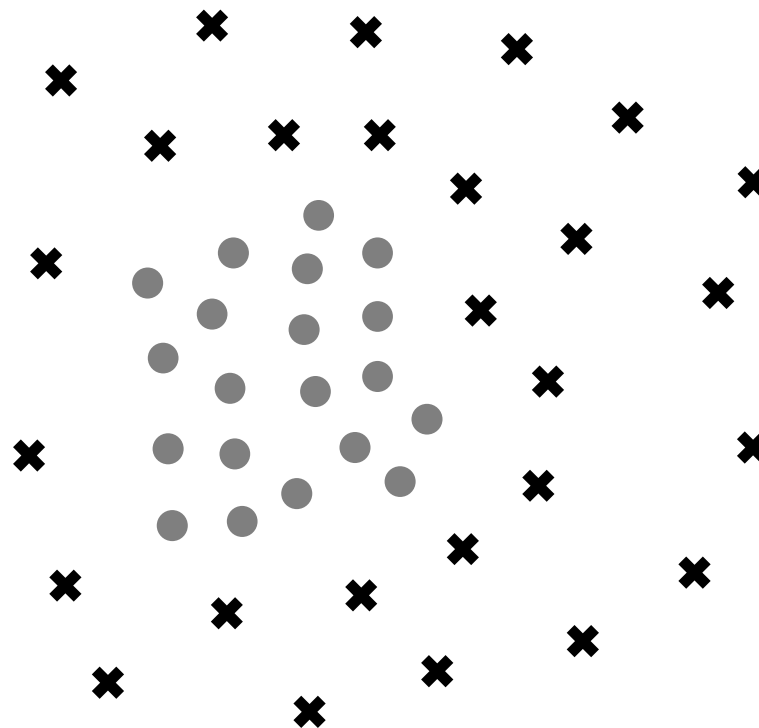
Near-OOD



Far-OOD



● : In-distribution  
✕ : Out-of-distribution



# Introduction (Out-of-Distribution Detection)

- Definitional Overlap in Out-of-Distribution Detection

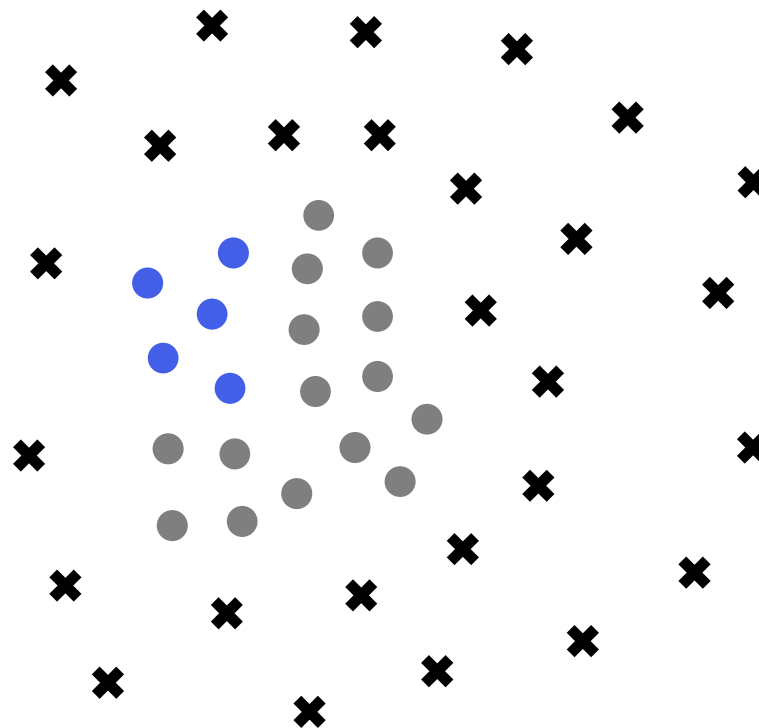
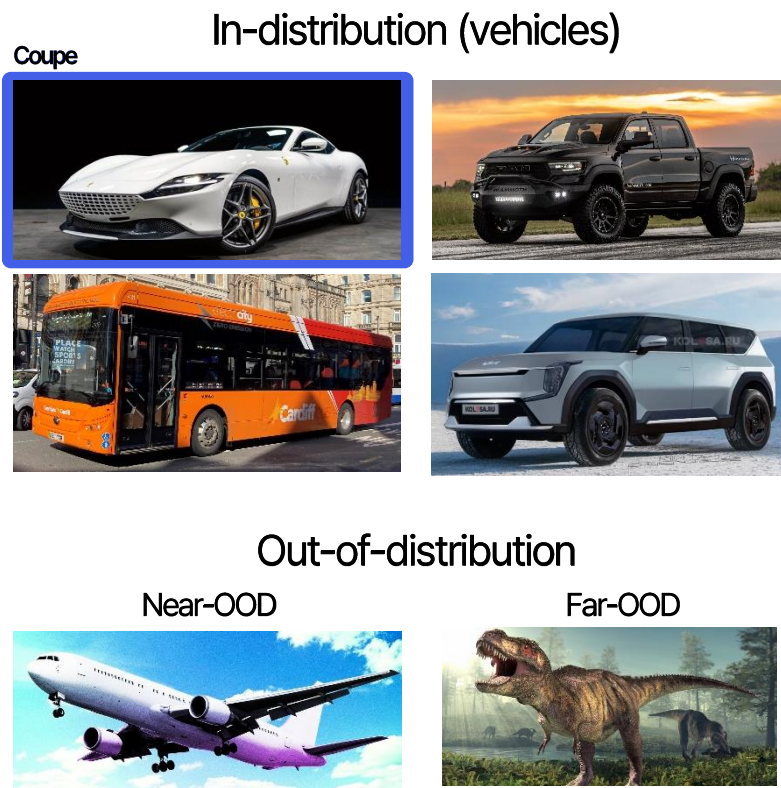
Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.

✕ : Out-of-distribution

● : In-distribution (coupe)



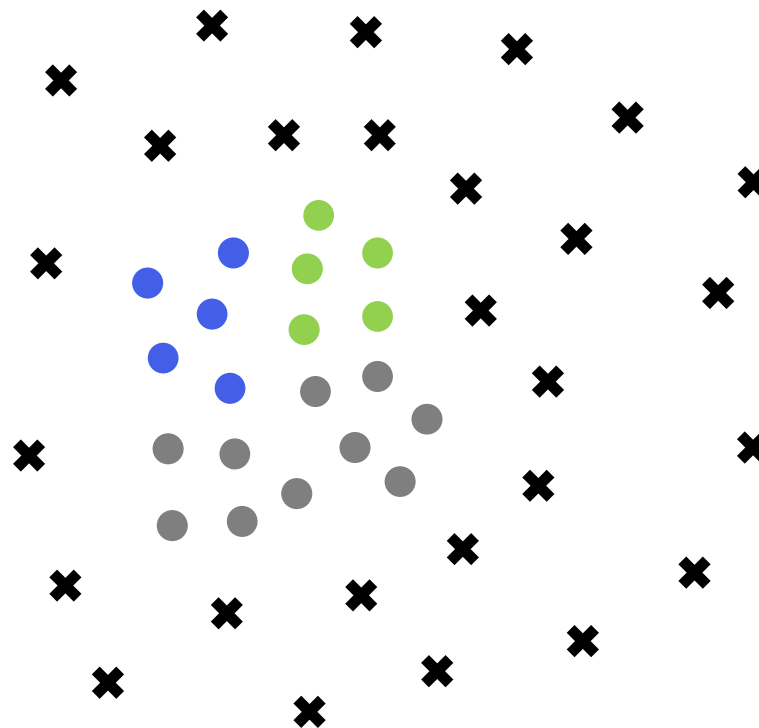
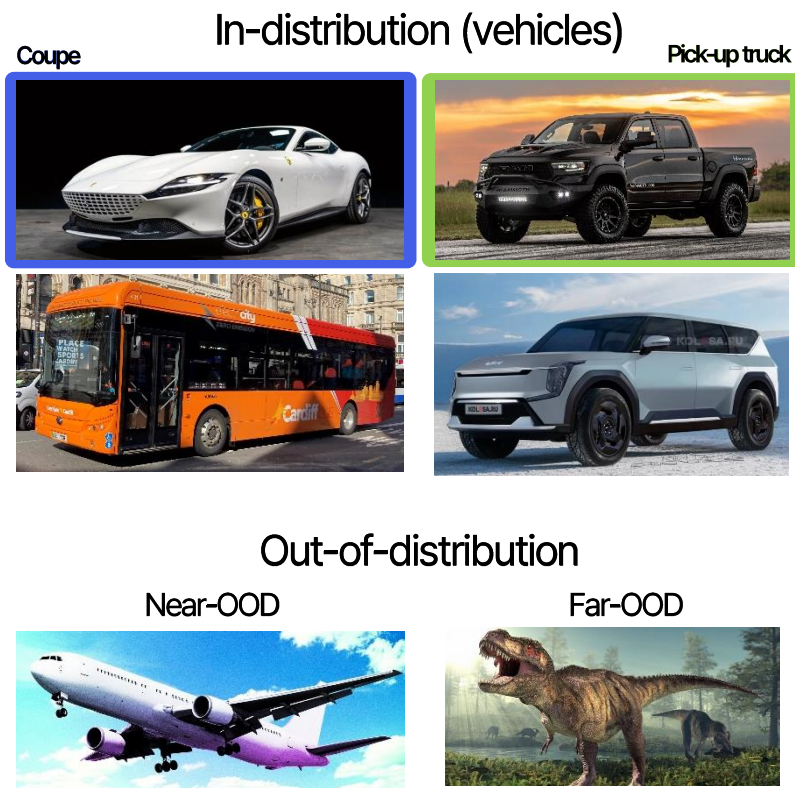
# Introduction (Out-of-Distribution Detection)

- Definitional Overlap in Out-of-Distribution Detection

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.



- ✕ : Out-of-distribution
- : In-distribution (coupe)
- : In-distribution (pick-up truck)



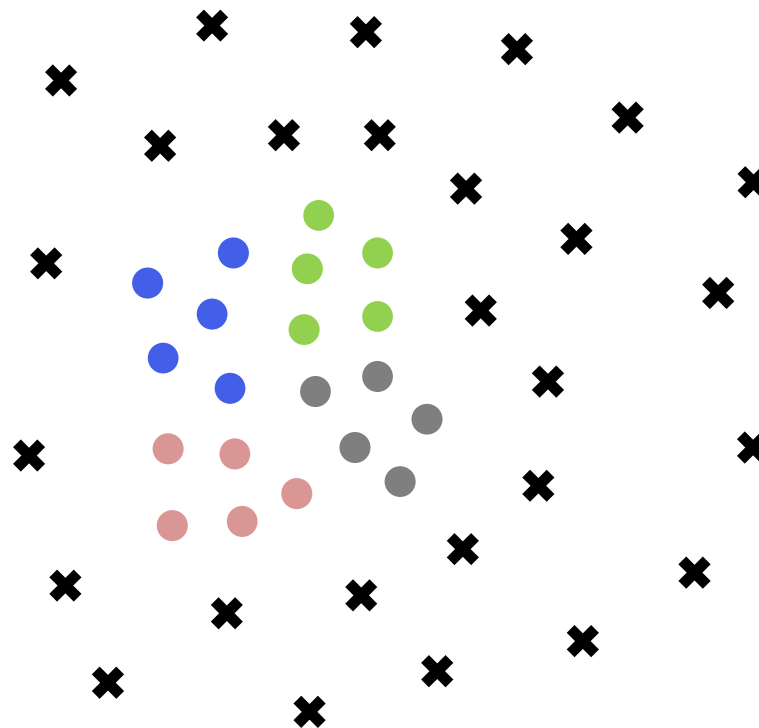
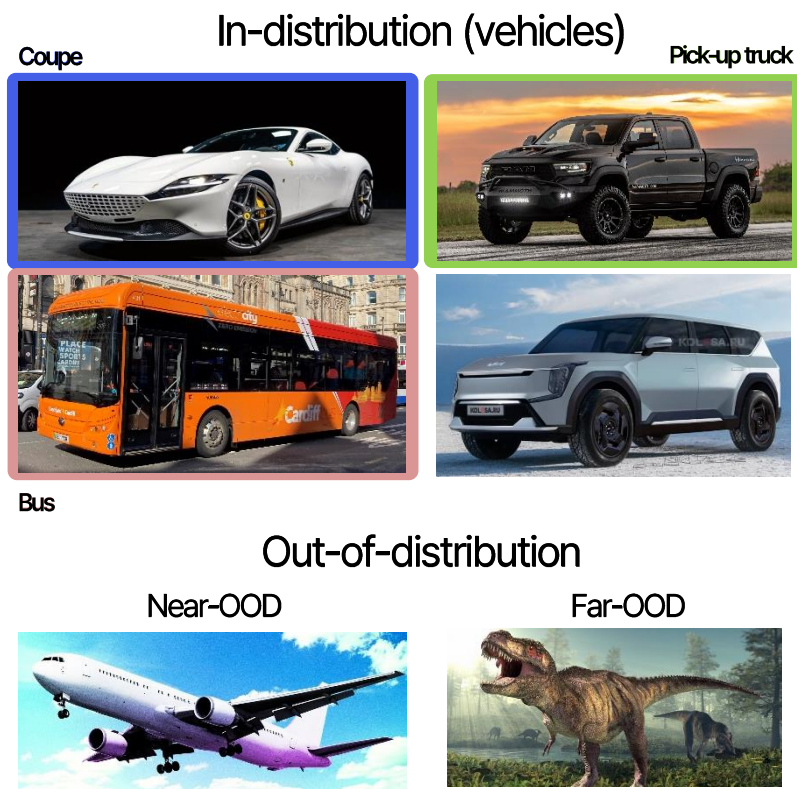
# Introduction (Out-of-Distribution Detection)

- Definitional Overlap in Out-of-Distribution Detection

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.



- ✕ : Out-of-distribution
- : In-distribution (coupe)
- : In-distribution (pick-up truck)
- : In-distribution (bus)

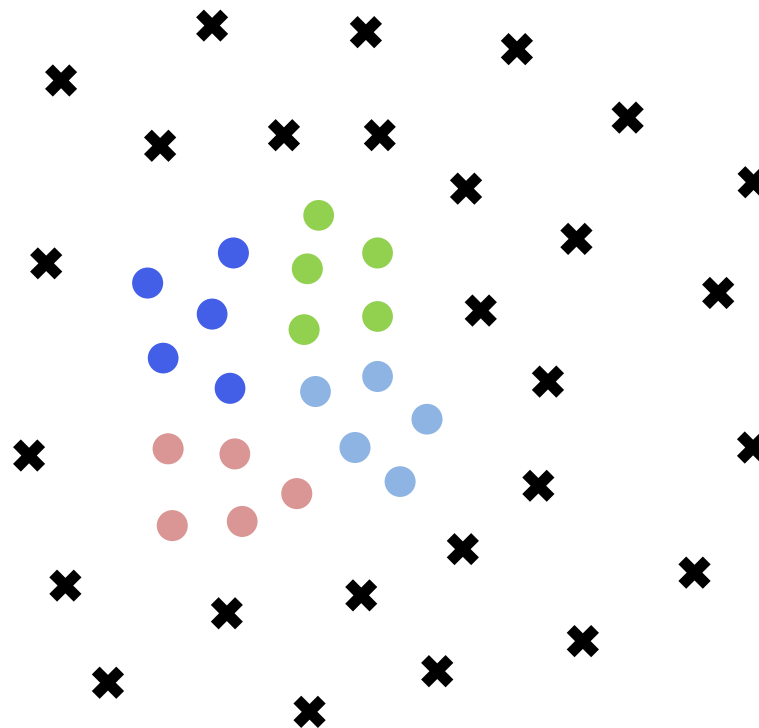
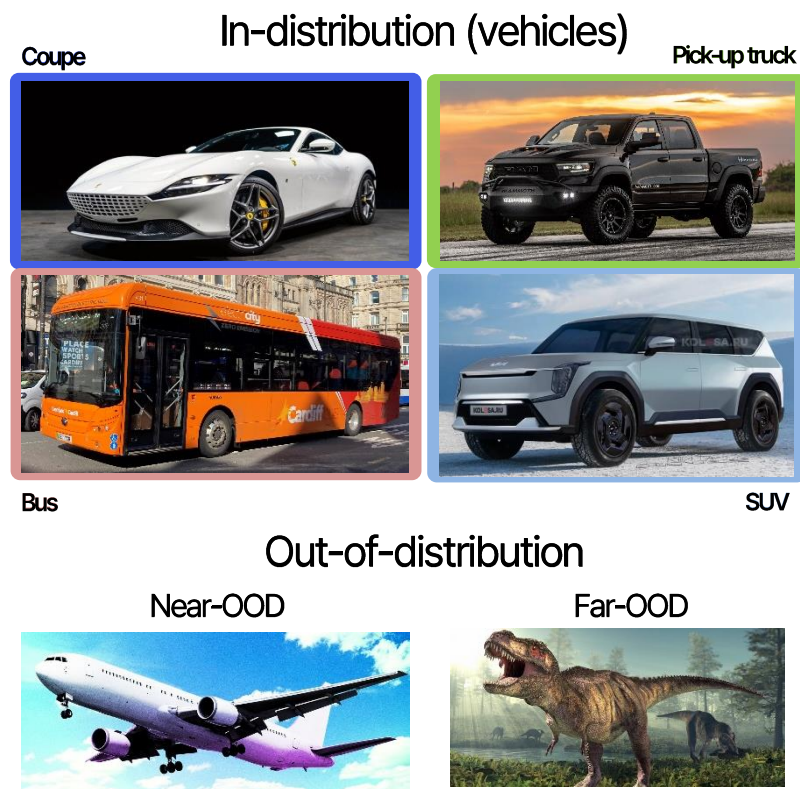
# Introduction (Out-of-Distribution Detection)

## • Definitional Overlap in Out-of-Distribution Detection

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.



- ✕ : Out-of-distribution
- : In-distribution (coupe)
- : In-distribution (pick-up truck)
- : In-distribution (bus)
- : In-distribution (SUV)

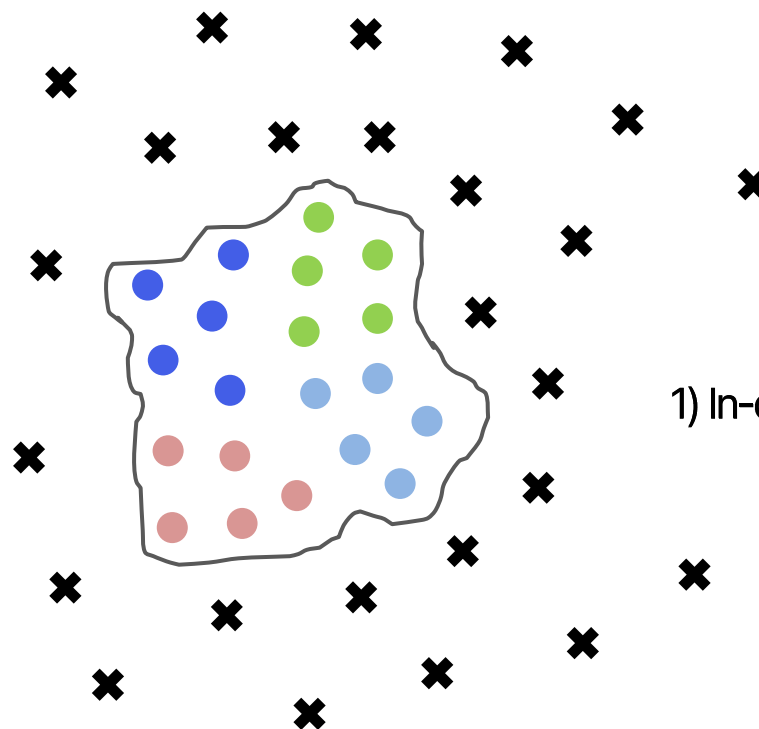
# Introduction (Out-of-Distribution Detection)

## • Definitional Overlap in Out-of-Distribution Detection

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.



- ✕ : Out-of-distribution
- : In-distribution (coupe)
- : In-distribution (pick-up truck)
- : In-distribution (bus)
- : In-distribution (SUV)

1) In-distribution 과 out-of-distribution 의 분포 차이 탐지

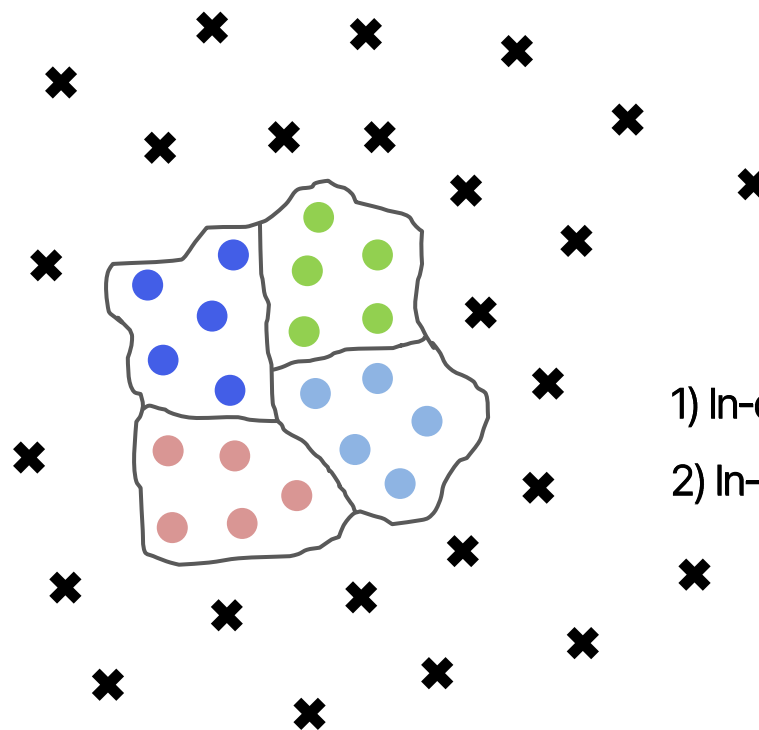
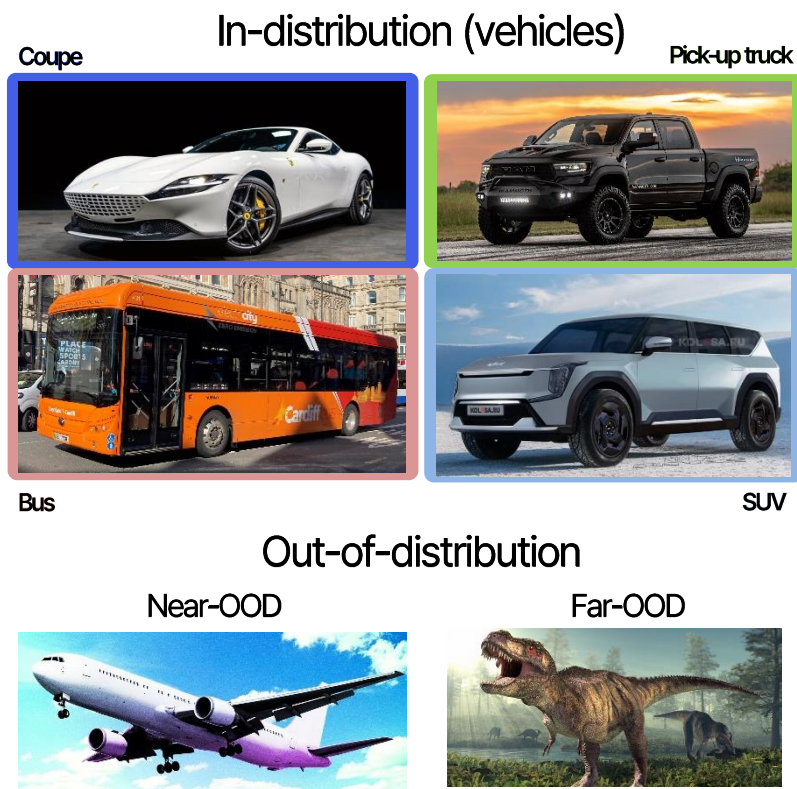
# Introduction (Out-of-Distribution Detection)

## • Definitional Overlap in Out-of-Distribution Detection

Anomaly detection, out-of-distribution detection, open-set recognition ...

Out-of-distribution(OOD) detection : in-distribution 과 다른 것들을 탐지해주는 모든 개념을 의미하며 논문들에서 혼용 표기되고 있음 [1]

Computer science 분야에서는 현실적으로 실험 세팅/실험 목표에 따라 분류하고 있음.



- ✕ : Out-of-distribution
- : In-distribution (coupe)
- : In-distribution (pick-up truck)
- : In-distribution (bus)
- : In-distribution (SUV)

## Open-set recognition

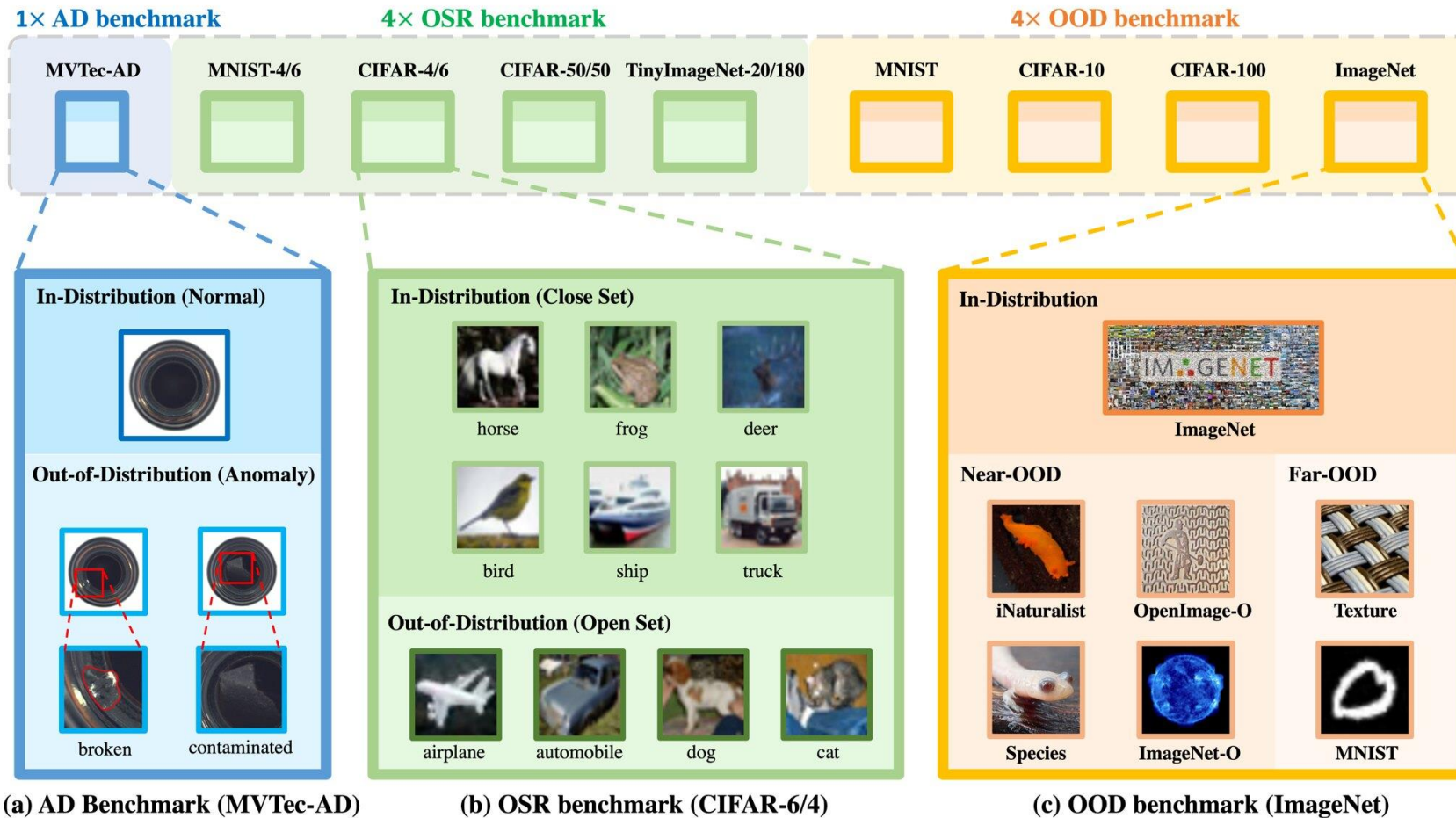
- 1) In-distribution 과 out-of-distribution 의 분포 차이 탐지
- 2) In-distribution 의 분류 성능 확보



# Introduction (Out-of-Distribution Detection)

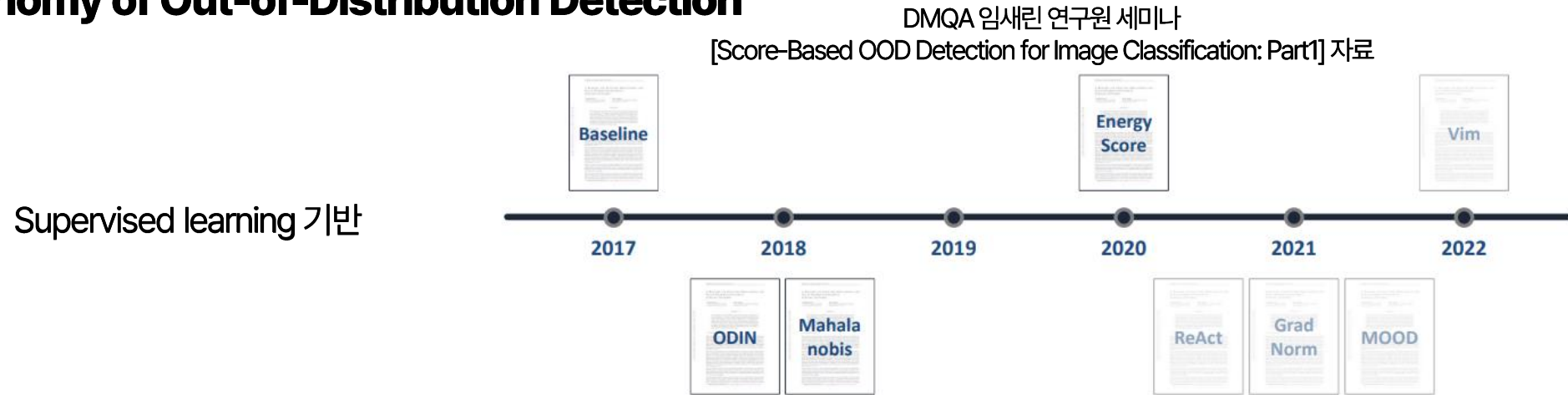
- Definitional Overlap in Out-of-Distribution Detection

Computer science 분야에서는 널리 쓰이는 OOD detection 실험 세팅은 아래와 같음.



# Introduction (Out-of-Distribution Detection)

- **Taxonomy of Out-of-Distribution Detection**



Unsupervised learning 기반

## Likelihood

- [Arxiv 2018] WAIC, but Why, Generative Ensembles for Robust Anomaly Detection
- [NeurIPS 2020] Why Normalizing Flows Fail to Detect Out-of-Distribution Data
- [AISTATS 2021] Density of States Estimation for Out-of-Distribution Detection

## Reconstruction-based

- Autoencoder-based
- GAN-based
- Diffusion-based

# Introduction (Out-of-Distribution Detection)

- **(Diffusion-based) Unsupervised Out-of-Distribution Detection**

[CVPR Workshop 2023] Denoising Diffusion Models for Out-of-Distribution Detection

[ICML 2023] Unsupervised Out-of-Distribution Detection with Diffusion Inpainting

[NeurIPS 2023] Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models

## 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 Tudosi  
King's College London  
petru.tudosi@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, 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. We validate our approach both on standard computer-vision datasets and on higher dimension medical datasets. Our approach outperforms not only reconstruction-based methods, but also state-of-the-art generative-based approaches. Code is available at <https://github.com/marksgraham/ddpm-ood>.

### 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

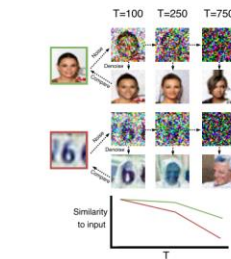


Figure 1. Reconstruction-based OOD detection, with the example of a model trained on CelebA. An in-distribution image from CelebA and an OOD image from SVHN are noised to various levels, reconstructed using the DDPM, and compared to the input. The similarity between inputs and reconstructions is plotted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

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.

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 [1, 9, 34]. A number of methods have been proposed to address these shortcomings.

## Unsupervised Out-of-Distribution Detection with Diffusion Inpainting

Zhenzhen Liu<sup>1</sup> Jin Peng Zhou<sup>1</sup> Yufan Wang<sup>1</sup> Kilian Q. Weinberger<sup>1</sup>

### Abstract

Unsupervised out-of-distribution detection (OOD) seeks to identify out-of-domain data by learning only from unlabeled in-domain data. We present a novel approach for this task – Lift, Map, Detect (LMD) – that leverages recent advancement in diffusion models. Diffusion models are one type of generative models. At their core, they learn an iterative denoising process that gradually maps a noisy image closer to their training manifolds. LMD leverages this intuition for OOD detection. Specifically, LMD lifts an image off its original manifold by corrupting it, and maps it towards the in-domain manifold with a diffusion model. For an out-of-domain image, the mapped image would have a large distance away from its original manifold, and LMD would identify it as OOD accordingly. We show through extensive experiments that LMD achieves competitive performance across a broad variety of datasets. Code can be found at [https://github.com/zhenzhen1/lift\\_map\\_detect](https://github.com/zhenzhen1/lift_map_detect).

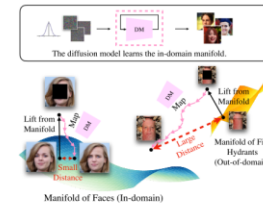


Figure 1. The pictorial intuition behind LMD for OOD detection. A diffusion model learns a mapping to the in-domain manifold. LMD lifts an image off its manifold by masking, and uses the diffusion model to move it towards the in-domain manifold. An in-domain image would have a much smaller distance between the original and mapped locations than its out-of-domain counterparts.

### 1. Introduction

Out-of-distribution (OOD) detection seeks to classify whether a data point belongs to a particular domain. It is especially important, because machine learning models typically assume that test-time samples are drawn from the same distribution as the training data. If the test data do not follow the training distribution, they can inadvertently produce non-sensical results. The increased use of machine learning models in high-stake areas, such as medicine [Hamet & Tremblay, 2017] and criminal justice [Rigano, 2019], amplifies the importance of OOD detection. For example, if a doctor mistakenly inputs a chest X-ray into a brain tumor detector, the model would likely still return a prediction –

<sup>1</sup>Equal contribution <sup>1</sup>Department of Computer Science, Cornell University, Ithaca, New York, USA. Correspondence to: Zhenzhen Liu <z353@cornell.edu>, Jin Peng Zhou <jz353@cornell.edu>.

Proceedings of the 10<sup>th</sup> International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

which would be meaningless and possibly misleading.

Previous researches have studied OOD detection under different settings: supervised and unsupervised. Within the supervised setup, the supervision can originate from different sources. In the most informed setting, one assumes access to representative out-of-domain samples. These allow one to train an OOD detector as a classifier distinguishing in-domain from out-of-domain data, and achieve high performance [Hendrycks et al., 2018; Ruff et al., 2019] – as long as the out-of-domain data do not deviate from the assumed out-of-domain distribution. In many practical applications, however, such knowledge is unattainable. In fact, out-of-domain data can be highly diverse and unpredictable.

A significantly more relaxed assumption is to only require access to an in-domain classifier or class labels. Under this setting, methods such as Hendrycks & Gimpel (2016); Liang et al. (2017); Lee et al. (2018); Huang et al. (2022); Wang et al. (2022) have achieved competitive performance. Although less informed, this setting relies on two implicit assumptions: the in-domain data have well-defined classes,

## Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models

Sungik Choi<sup>1</sup> Hankook Lee<sup>1</sup> Honglak Lee<sup>1</sup> Moontae Lee<sup>1,2</sup>  
<sup>1</sup>LG AI Research <sup>2</sup>University of Illinois Chicago  
{sungik.choi, hankook.lee, honglak.lee, moontae.lee}@lgresearch.ai

### Abstract

Novelty detection is a fundamental task of machine learning which aims to detect abnormal (i.e. out-of-distribution (OOD)) samples. Since diffusion models have recently emerged as the de facto standard generative framework with surprising generation results, novelty detection via diffusion models has also gained much attention. Recent methods have mainly utilized the reconstruction property of in-distribution samples. However, they often suffer from detecting OOD samples that share similar background information to the in-distribution data. Based on our observation that diffusion models can project any sample to an in-distribution sample with similar background information, we propose *Projection Regret (PR)*, an efficient novelty detection method that mitigates the bias of non-semantic information. To be specific, PR computes the perceptual distance between the test image and its diffusion-based projection to detect abnormality. Since the perceptual distance often fails to capture semantic changes when the background information is dominant, we cancel out the background bias by comparing it against recursive projections. Extensive experiments demonstrate that PR outperforms the prior art of generative-model-based novelty detection methods by a significant margin.

### 1 Introduction

Novelty detection [1], also known as out-of-distribution (OOD) detection, is a fundamental machine learning problem, which aims to detect abnormal samples drawn from far from the training distribution (i.e., in-distribution). This plays a vital role in many deep learning applications because the behavior of deep neural networks on OOD samples is often unpredictable and can lead to erroneous decisions [2]. Hence, the detection ability is crucial for ensuring reliability and safety in practical applications, including medical diagnosis [3], autonomous driving [4], and forecasting [5].

The principled way to identify whether a test sample is drawn from the training distribution  $p_{\text{data}}(\mathbf{x})$  or not is to utilize an explicit or implicit generative model. For example, one may utilize the likelihood function directly [6] or its variants [7, 8] as an OOD detector. Another direction is to utilize the generation ability, for example, reconstruction loss [9] or gradient representations of the reconstruction loss [10]. It is worth noting that the research direction of utilizing generative models for out-of-distribution detection has become increasingly important in recent years as generative models have been successful across various domains, e.g., vision [11, 12] and language [13], but also they have raised various social issues including deepfake [14] and hallucination [15].

Among various generative frameworks, diffusion models have recently emerged as the most popular framework due to their strong generation performance, e.g., high-resolution images [16], and its wide applicability, e.g., text-to-image synthesis [11]. Hence, they have also gained much attention as an attractive tool for novelty detection. For example, Mahmood et al. [17] and Liu et al. [18] utilize the reconstruction error as the OOD detection metric using diffusion models with Gaussian noises [17] or checkerboard masking [18]. Somewhat unexpectedly, despite the high-quality generation results,

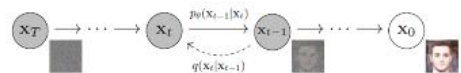
37th Conference on Neural Information Processing Systems (NeurIPS 2023).

# Introduction (Out-of-Distribution Detection)


- DMQA Seminar (Diffusion)

**종료** Diffusion Probabilistic Models (DDPM)

- Forward process: 데이터( $x_0$ ) + 노이즈  $\rightarrow$  latent 노이즈( $z_T$ )
- Reverse process: latent 노이즈( $z_T$ ) + 노이즈 제거  $\rightarrow$  데이터( $x_0$ )
- 노이즈를 제거하는 reverse process를 학습할 수 있다면 latent 노이즈로부터 데이터 생성 가능



Score-based Generative Models and Diffu

발표자:  조한샘

📅 2022년 2월 11일  
🕒 오후 1시 ~  
📺 온라인 비디오 시청 (YouTube)

세미나 정보 보기 →

**종료** Improving Sampling Speed of Diffusion Models

Open DMQA Seminar  
2023.02.10

조한샘


Improving Sampling Speed of Diffusion M

발표자:  조한샘

📅 2023년 2월 10일  
🕒 오후 1시 ~  
📺 온라인 비디오 시청 (YouTube)


세미나 정보 보기 →

**종료** Conditional Diffusion Models



Jong Hyun Lee  
2023.06.09

Conditional Diffusion Models

발표자:  이종현

📅 2023년 6월 16일  
🕒 오전 12시 ~  
📺 온라인 비디오 시청 (YouTube)

세미나 정보 보기 →



# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

방법: ① In-distribution 으로 학습된 diffusion model 을 준비 (아래 예제에서는 CelebA 로 학습)

Diffusion model  
trained with  
**CelebA** (in-distribution)

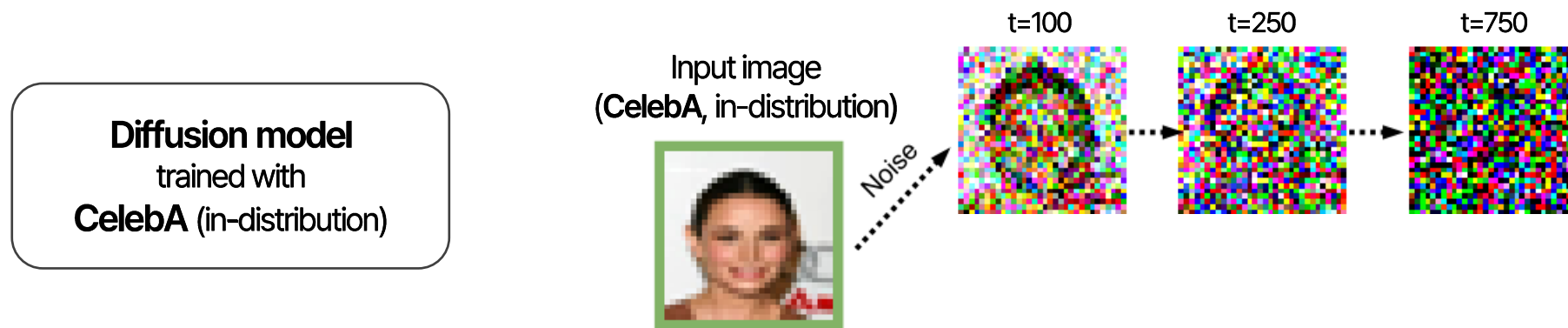
# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

- 방법:
- ① In-distribution 으로 학습된 diffusion model 을 준비 (아래 예제에서는 CelebA 로 학습)
  - ② 분석을 원하는 image 에 gaussian noise 를  $t=0 \sim 1000$  scale 로 넣어줌  $\rightarrow q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha})\mathbf{I})$



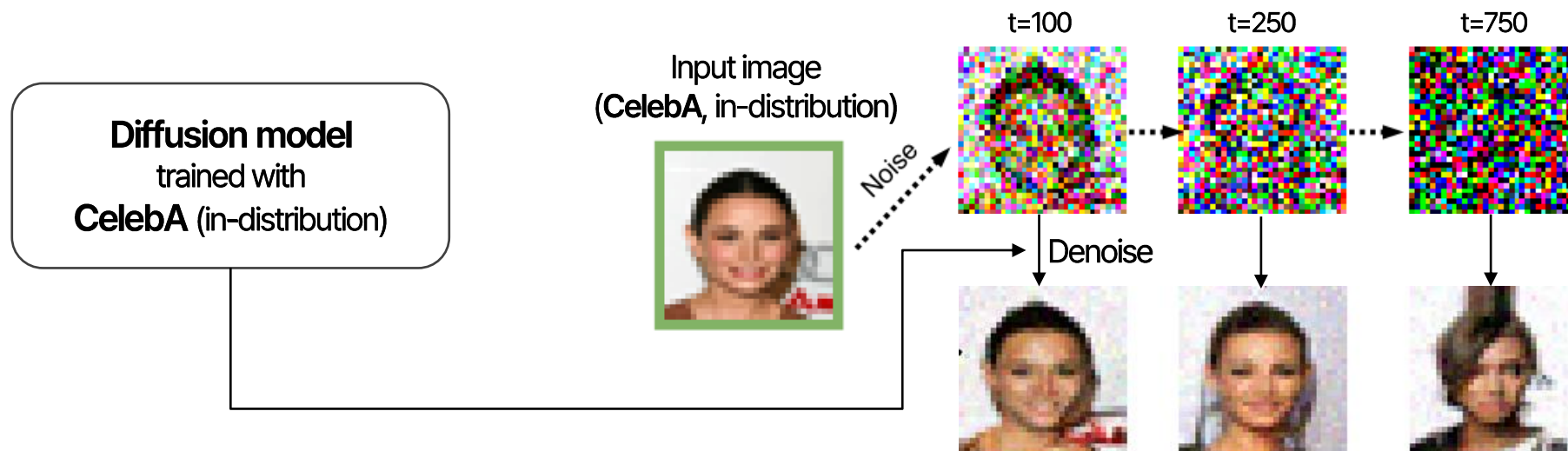
# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

- 방법:
- ① In-distribution 으로 학습된 diffusion model 을 준비 (아래 예제에서는 CelebA 로 학습)
  - ② 분석을 원하는 image 에 gaussian noise 를  $t=0 \sim 1000$  scale 로 넣어줌  $\rightarrow q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha})\mathbf{I})$
  - ③ 사용자가 선택한  $t$  step ( $t=10, 20, \dots, 990$ ) 에 대해 준비한 diffusion model 로 denoising 수행



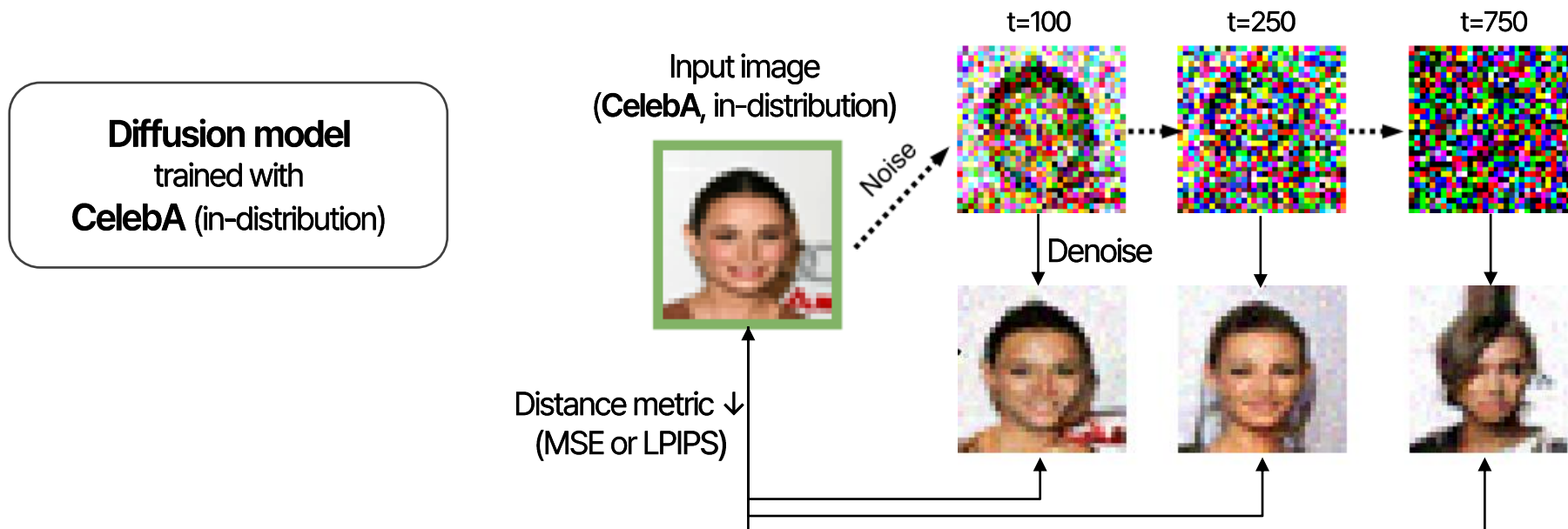
# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

- 방법:
- ① In-distribution 으로 학습된 diffusion model 을 준비 (아래 예제에서는 CelebA 로 학습)
  - ② 분석을 원하는 image 에 gaussian noise 를  $t=0 \sim 1000$  scale 로 넣어줌  $\rightarrow q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha})\mathbf{I})$
  - ③ 사용자가 선택한  $t$  step ( $t=10, 20, \dots, 990$ ) 에 대해 준비한 diffusion model 로 denoising 수행
  - ④ 원본 image 와 reconstruction image 를 distance metric (MSE or LPIPS) 으로 계산하여 그 값을 OOD 지표로 사용



# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

## LPIPS (Learned Perceptual Image Patch Similarity)

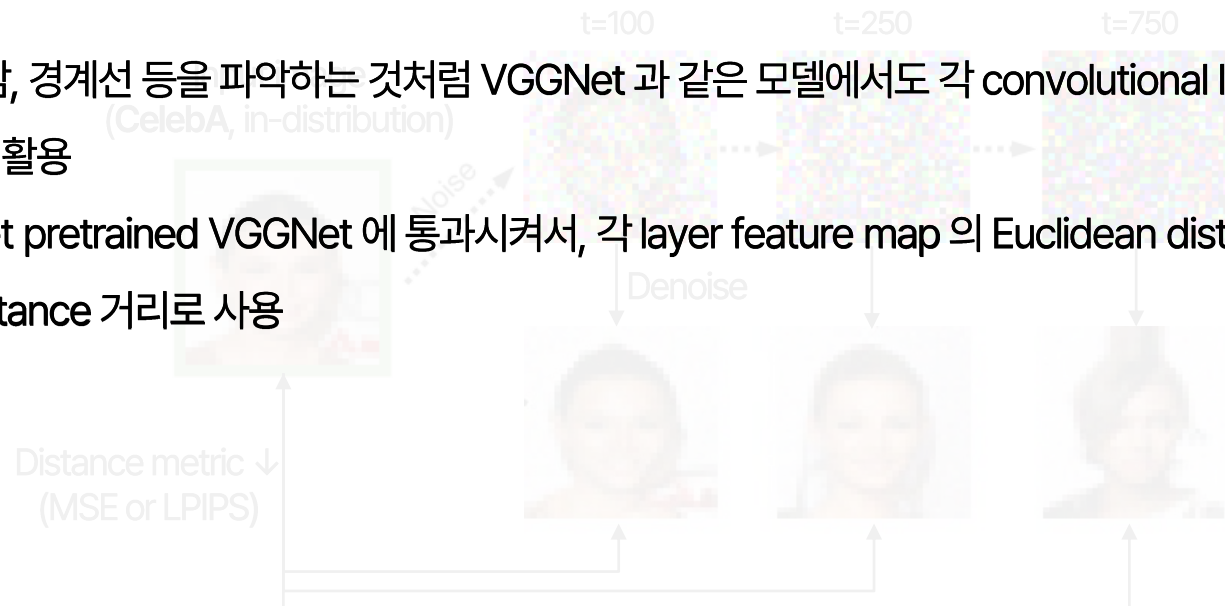
방법: ① In-distribution 으로 학습된 diffusion model 을 준비 (이제 예제에서는 CelebA 도 학습)

목표: 이미지 간의 distance 를 계산할 때, 인간의 인지적 특성을 반영하여 계산하도록 함

→ 실제로 사람이 느끼는 이미지 간의 유사도를 파악하고자 함

방법:

- ① 사람이 물체를 인식할 때, 형태, 질감, 경계선 등을 파악하는 것처럼 VGGNet 과 같은 모델에서도 각 convolutional layer 가 형태, 질감, 경계선 등을 파악할 수 있다는 점을 활용
- ② 두 개의 이미지를 하나의 ImageNet pretrained VGGNet 에 통과시켜서, 각 layer feature map 의 Euclidean distance 를 계산하여 평균내서 최종 score 를 LPIPS distance 거리로 사용



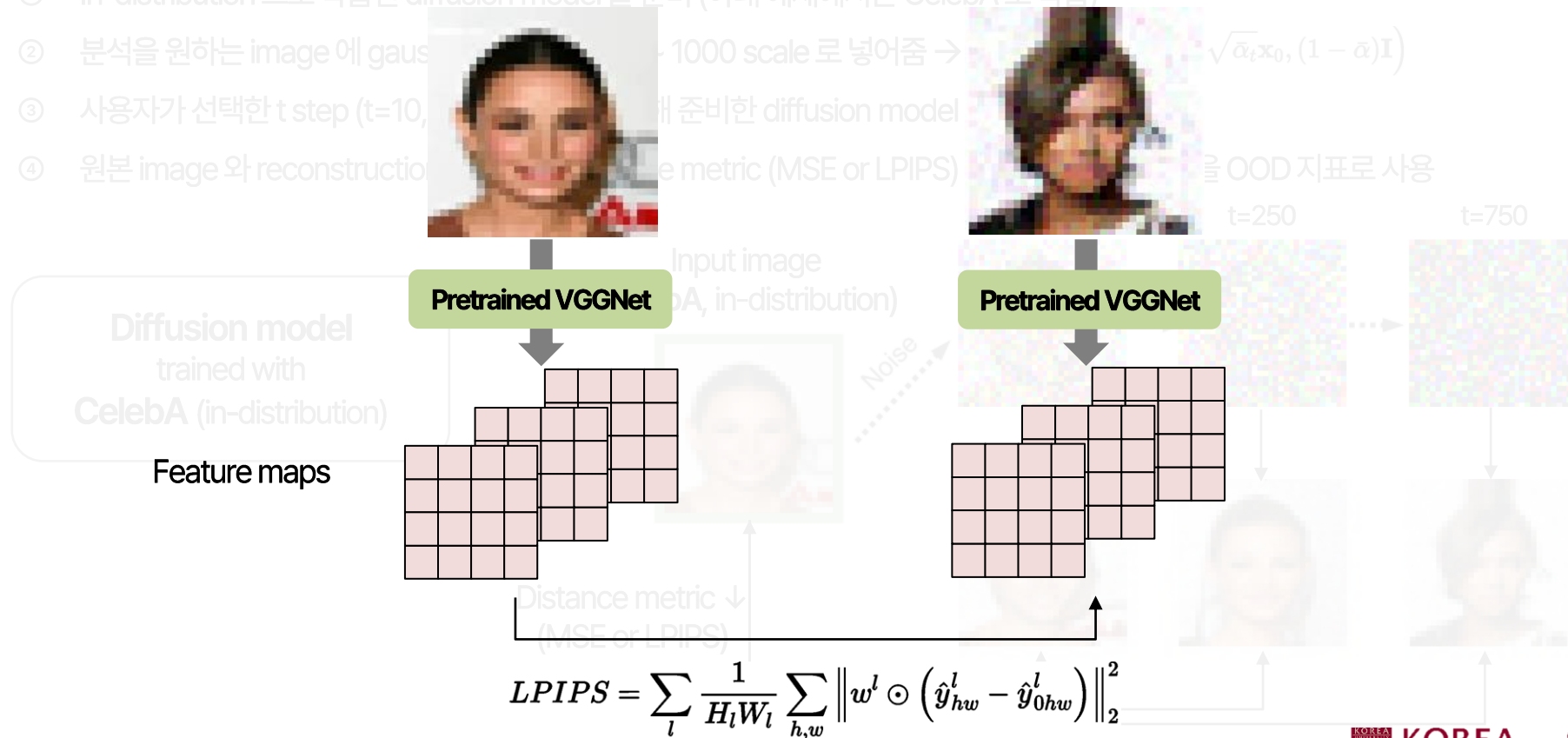
# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

방법: ① In-distribution 으로 학습된 diffusion model 을 준비 (아래 예제에서는 CelebA 로 학습)  
② 분석을 원하는 image 에 gaussian noise 를 1000 scale 로 넣어줌  $\rightarrow \sqrt{\alpha_t}x_0, (1 - \alpha)I$   
③ 사용자가 선택한 t step (t=10, 250, 750) 에 준비한 diffusion model 을 사용  
④ 원본 image 와 reconstruction image 에 distance metric (MSE or LPIPS) 을 OOD 지표로 사용





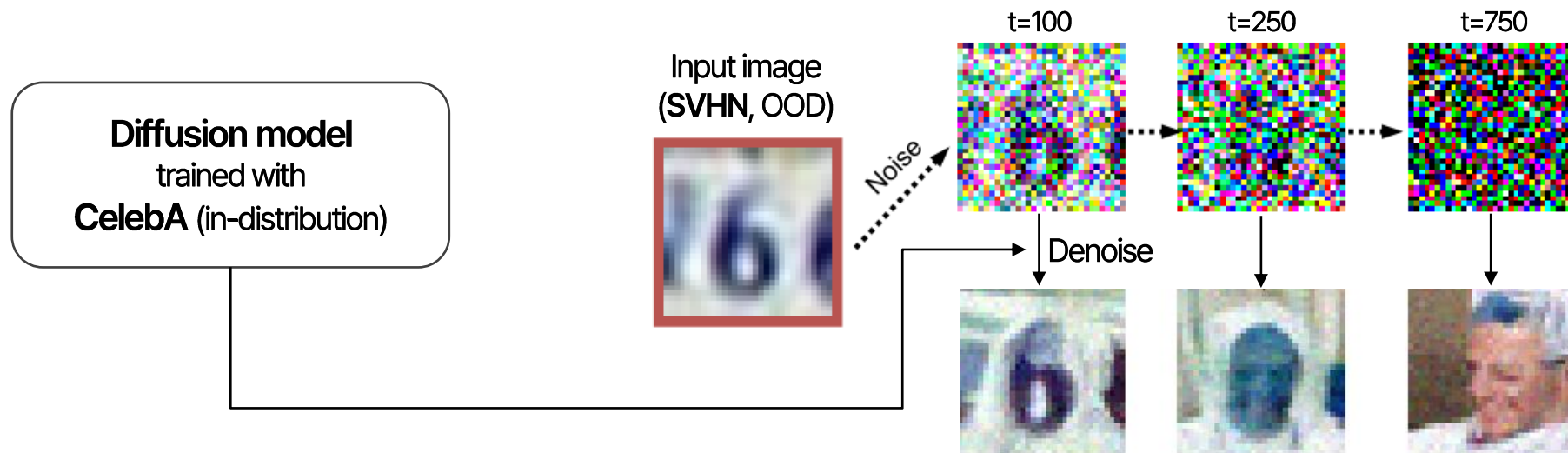
# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

- 방법:
- ① In-distribution 으로 학습된 diffusion model 을 준비 (아래 예제에서는 CelebA 로 학습)
  - ② 분석을 원하는 image 에 gaussian noise 를  $t=0 \sim 1000$  scale 로 넣어줌  $\rightarrow q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha})\mathbf{I})$
  - ③ 사용자가 선택한  $t$  step ( $t=10, 20, \dots, 990$ ) 에 대해 준비한 diffusion model 로 denoising 수행
  - ④ 원본 image 와 reconstruction image 를 distance metric (MSE or LPIPS) 으로 계산하여 그 값을 OOD 지표로 사용



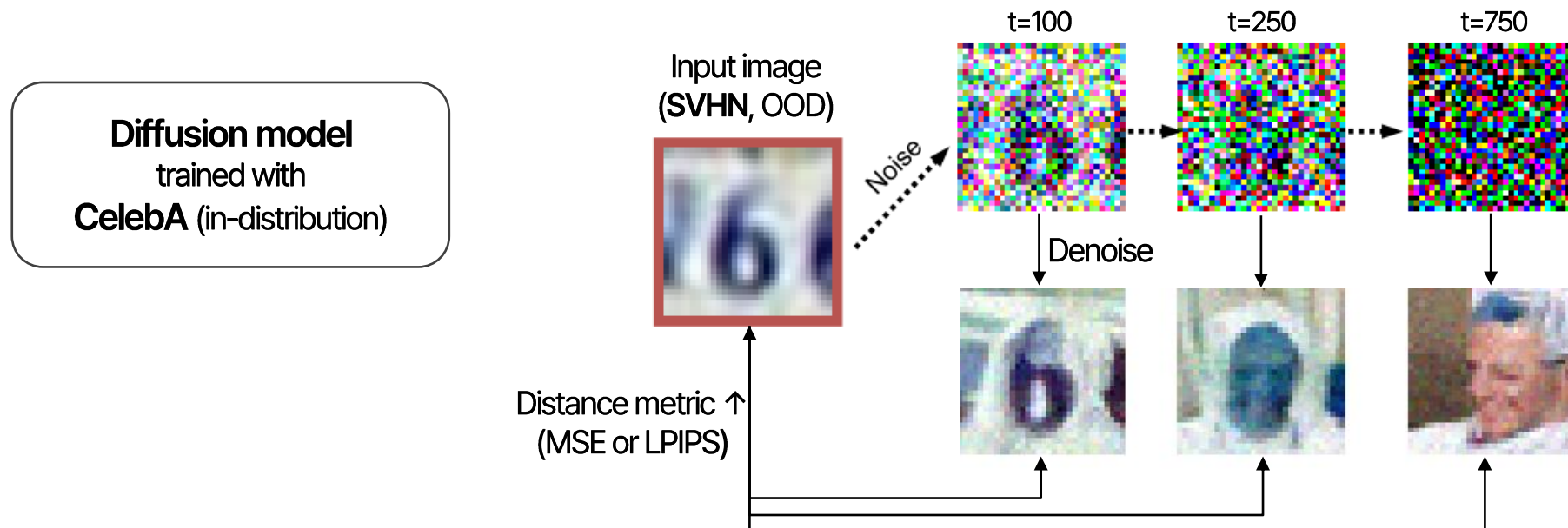
# Methods

- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (reconstruction) 에서 out-of-distribution data는 잘 안될 것

- 방법:
- ① In-distribution 으로 학습된 diffusion model 을 준비 (아래 예제에서는 CelebA 로 학습)
  - ② 분석을 원하는 image 에 gaussian noise 를  $t=0 \sim 1000$  scale 로 넣어줌  $\rightarrow q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha})\mathbf{I})$
  - ③ 사용자가 선택한  $t$  step ( $t=10, 20, \dots, 990$ ) 에 대해 준비한 diffusion model 로 denoising 수행
  - ④ 원본 image 와 reconstruction image 를 distance metric (MSE or LPIPS) 으로 계산하여 그 값을 OOD 지표로 사용



# Methods

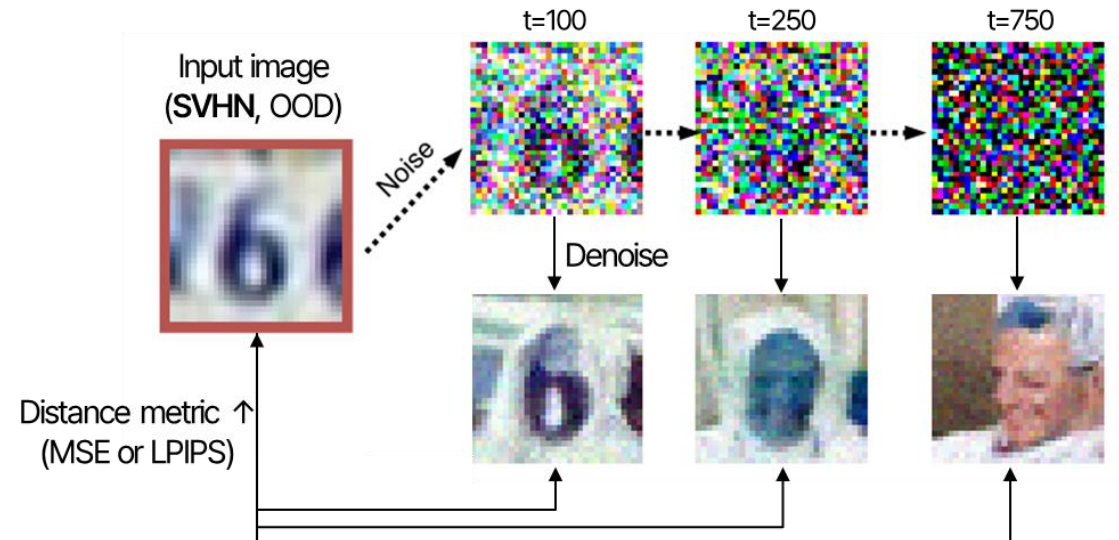
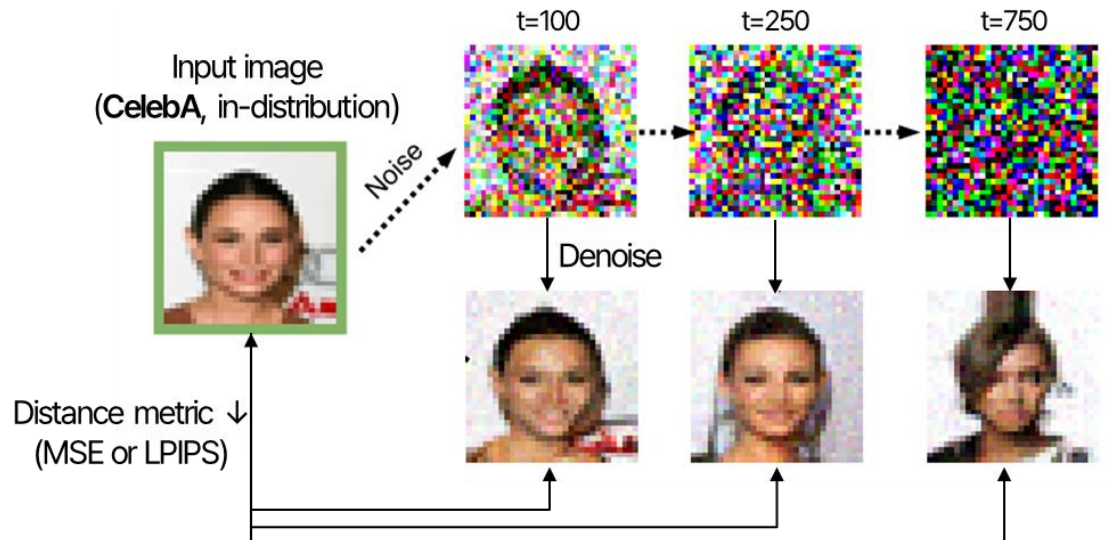
- [CVPR Workshop 2023] **Denoising Diffusion Models for Out-of-Distribution Detection**

기여점:

- ① Diffusion model 을 사용해서 최초로 OOD detection 을 수행함
- ② Likelihood, autoencoder 계열 알고리즘보다 성능이 우수함

단점:

- ① 빠른 PLMS sampler 을 사용하긴 하지만, 많은 t step 에 대해 전부 reconstruction 을 수행하여 시간 소요 (어떤 t 에서 변화할 지 모르므로)



# Methods

- [ICML 2023] **Unsupervised Out-of-Distribution Detection with Diffusion Inpainting**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

가정: in-distribution 으로 학습된 diffusion model 이 있다고 하면, 특정 task (inpainting) 에서 out-of-distribution data는 잘 안될 것



[ Score-based Generative Modeling Through SDEs 논문 내 inpainting 예시 ]

# Methods

- [ICML 2023] **Unsupervised Out-of-Distribution Detection with Diffusion Inpainting**

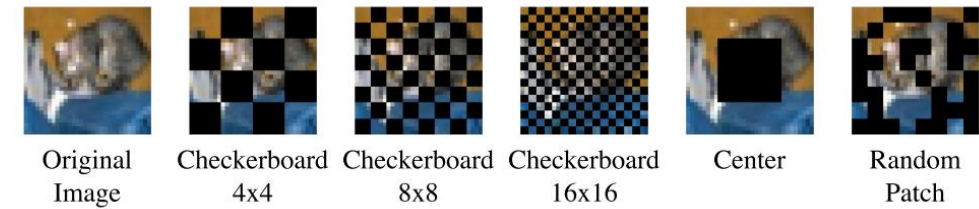
방법:

- ① In-distribution 으로 학습된 diffusion model 준비

**Diffusion model**  
trained with  
**MNIST** (in-distribution)



# Methods



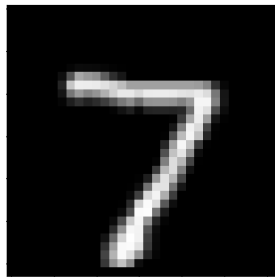
- [ICML 2023] **Unsupervised Out-of-Distribution Detection with Diffusion Inpainting**

방법:

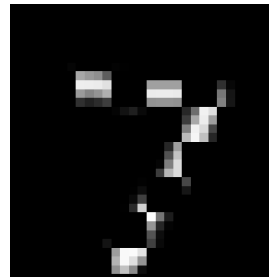
- ① In-distribution 으로 학습된 diffusion model 준비
- ② 분석을 원하는 image 에 masking 진행

Diffusion model  
trained with  
**MNIST** (in-distribution)

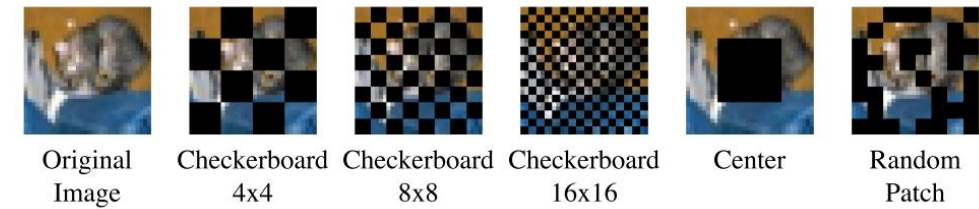
Input image  
(MNIST, in-distribution)



Masking  
(checkerboard 8x8)



# Methods

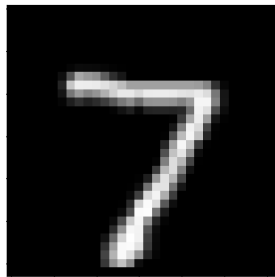


## • [ICML 2023] **Unsupervised Out-of-Distribution Detection with Diffusion Inpainting**

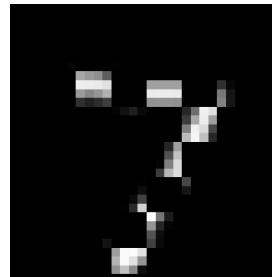
방법:

- ① In-distribution 으로 학습된 diffusion model 준비
- ② 분석을 원하는 image 에 masking 진행
- ③ Masking 완료된 image 에 준비한 diffusion model 을 통해 inpainting 진행

Input image  
(MNIST, in-distribution)

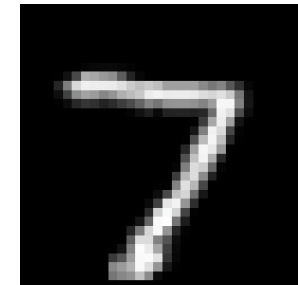


Masking  
(checkerboard 8x8)



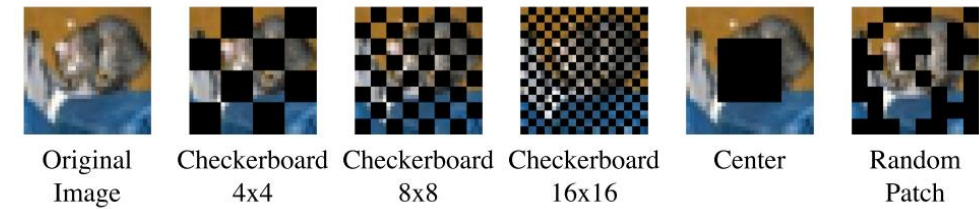
inpainting

Reconstruction image



Diffusion model  
trained with  
MNIST (in-distribution)

# Methods

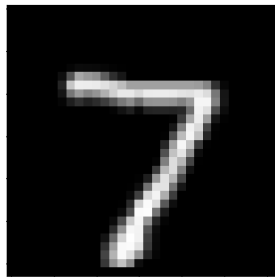


## • [ICML 2023] **Unsupervised Out-of-Distribution Detection with Diffusion Inpainting**

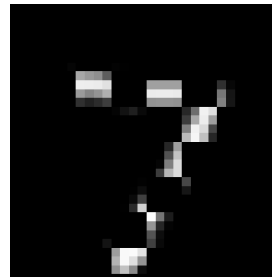
방법:

- ① In-distribution 으로 학습된 diffusion model 준비
- ② 분석을 원하는 image 에 masking 진행
- ③ Masking 완료된 image 에 준비한 diffusion model 을 통해 inpainting 진행
- ④ Input image 와 reconstruction image 의 distance metric 을 사용하여 OOD 지표로 활용

Input image  
(MNIST, in-distribution)

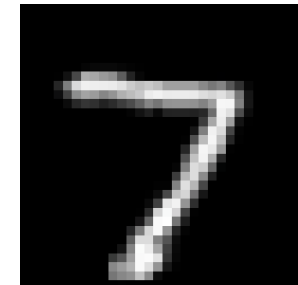


Masking  
(checkerboard 8x8)



inpainting

Reconstruction image

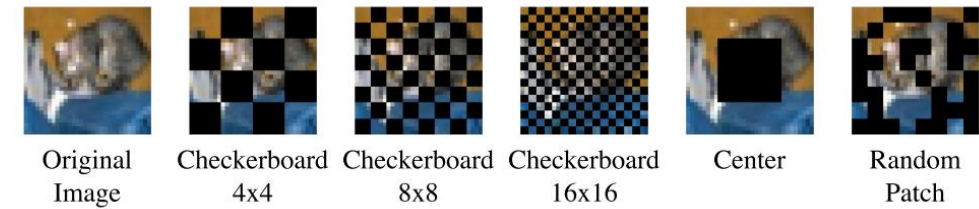


Distance metric ↓  
(LPIPS)

Diffusion model  
trained with  
MNIST (in-distribution)



# Methods

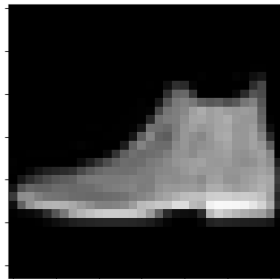


## • [ICML 2023] **Unsupervised Out-of-Distribution Detection with Diffusion Inpainting**

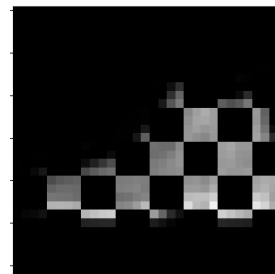
방법:

- ① In-distribution 으로 학습된 diffusion model 준비
- ② 분석을 원하는 image 에 masking 진행
- ③ Masking 완료된 image 에 준비한 diffusion model 을 통해 inpainting 진행
- ④ Input image 와 reconstruction image 의 distance metric 을 사용하여 OOD 지표로 활용

Input image  
(FashionMNIST, OOD)

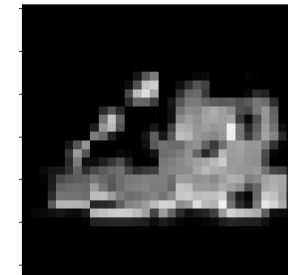


Masking  
(checkerboard 8x8)



inpainting

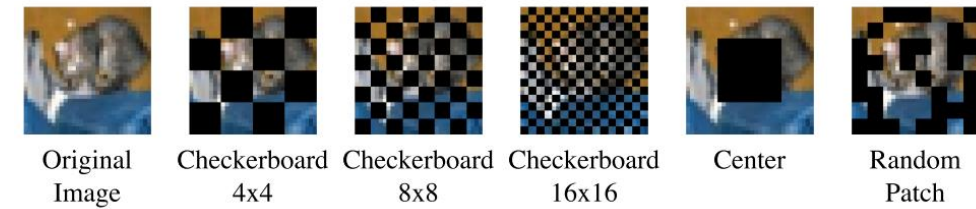
Reconstruction image



Distance metric ↑  
(LPIPS)

Diffusion model  
trained with  
**MNIST (in-distribution)**

# Methods



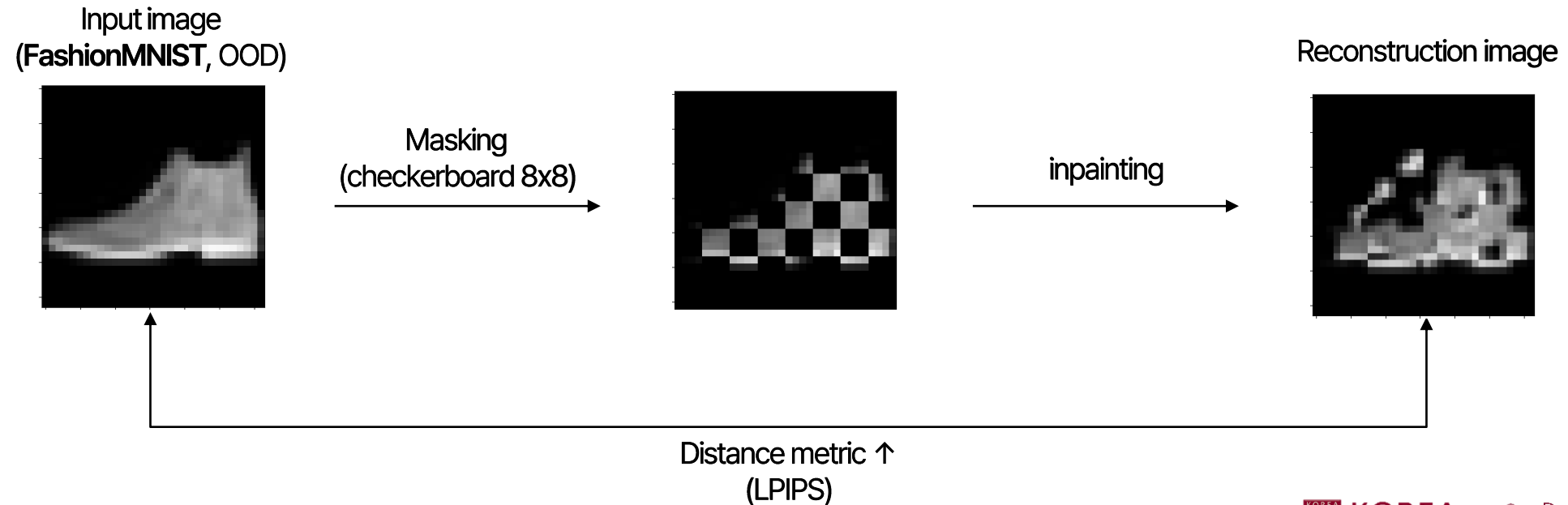
- [ICML 2023] **Unsupervised Out-of-Distribution Detection with Diffusion Inpainting**

## 기여점:

- ① Diffusion model 을 사용해서 최초로 OOD detection 을 수행함 (concurrent work)
- ② Likelihood, autoencoder 계열 알고리즘보다 성능이 우수함

## 단점:

- ① 어떤 masking 방법을 사용할 지 사용자가 선택해야 하는 단점이 있음
- ② Inference 속도가 다른 diffusion model 과 마찬가지로 오래 걸림.

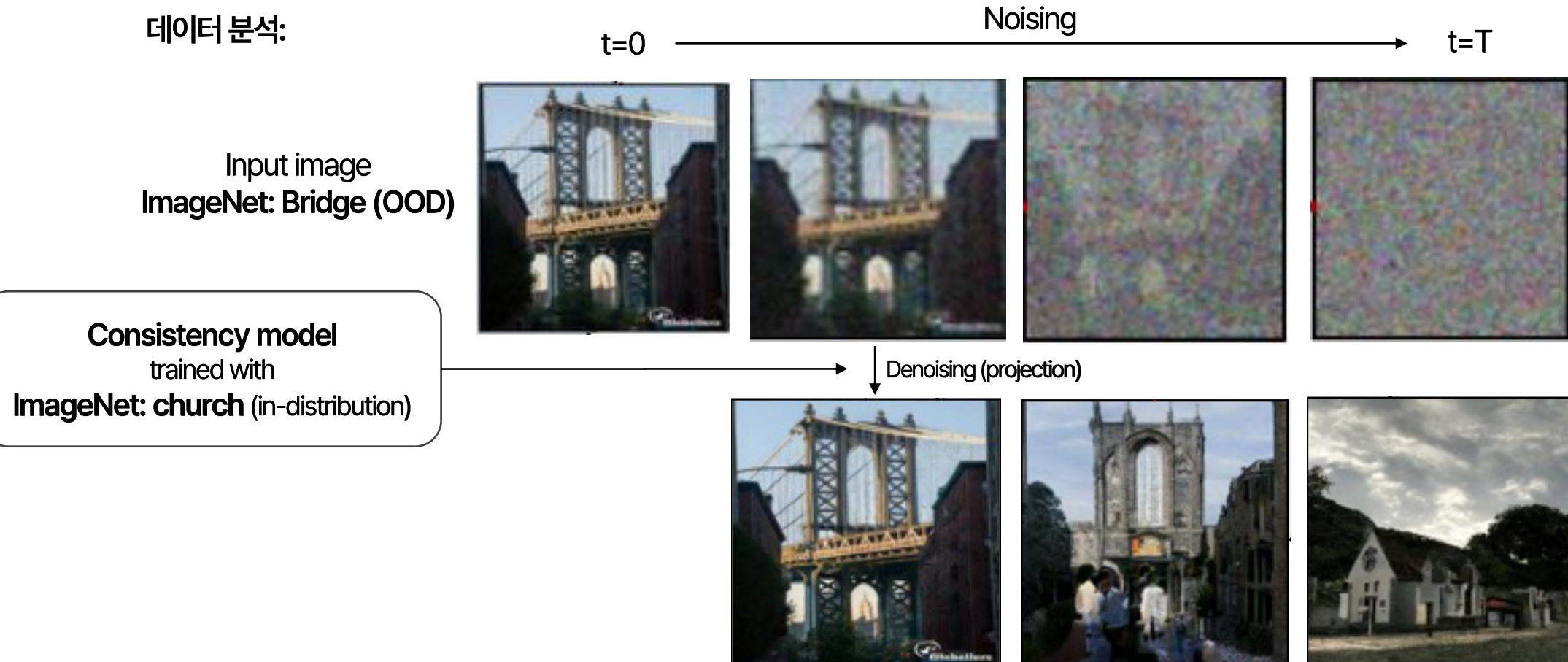


# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

데이터 분석:

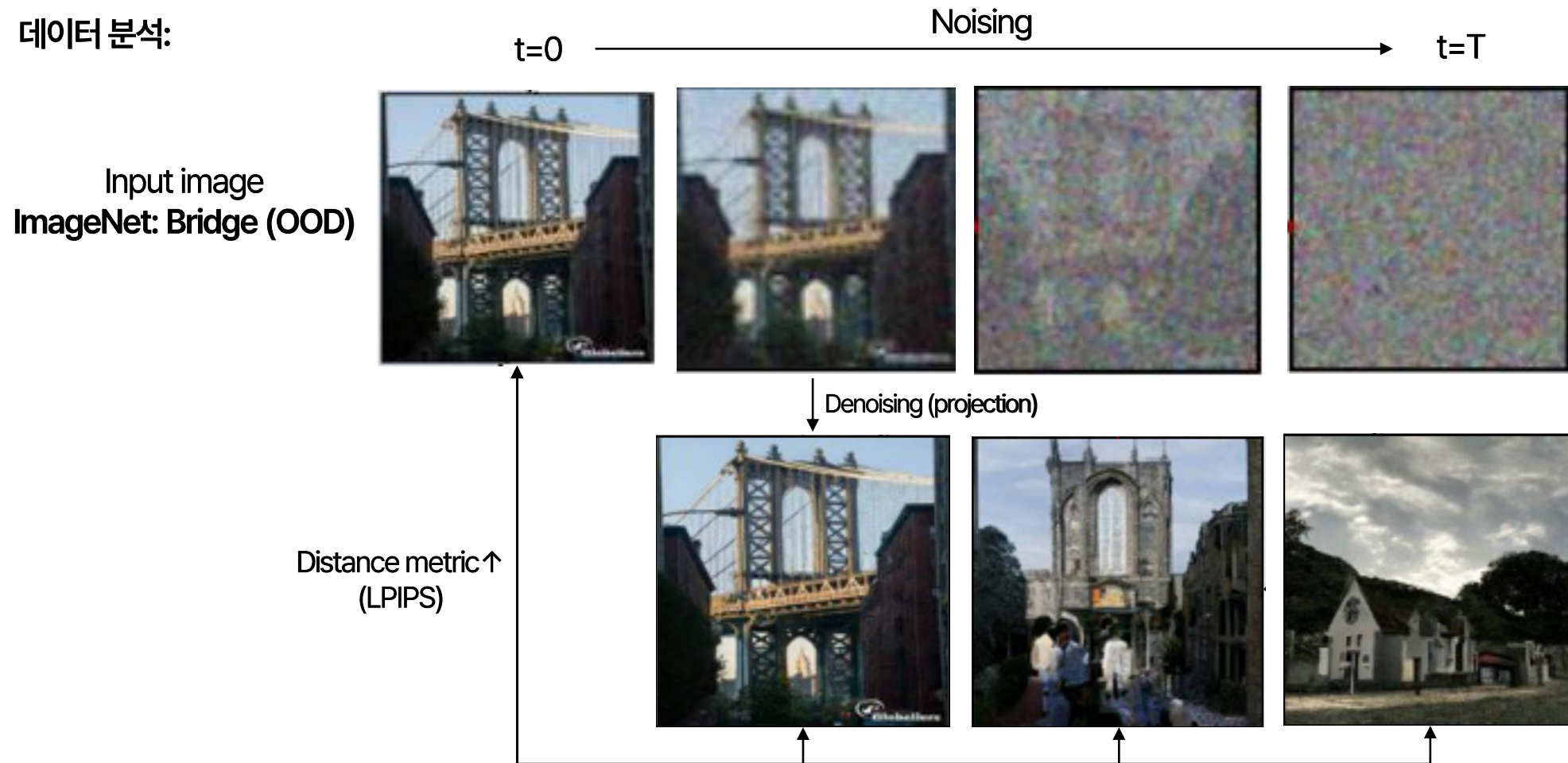


# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

데이터 분석:





# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

목표: in-distribution 과 out-of-distribution 의 차이를 정량화해서 OOD detection 을 수행하고자 함.

데이터 분석:

t=0  $\xrightarrow{\text{Noising}}$  t=T

Input image  
ImageNet: Bridge (OOD)



Church (in-distribution) 로 바뀌었지만, 배경 정보가 너무 유사하여 LPIPS metric 이 제대로 탐지하기 어려움.

Distance metric  $\uparrow$   
(LPIPS)

# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

방법:

- ① In-distribution 으로 학습된 consistency model 준비

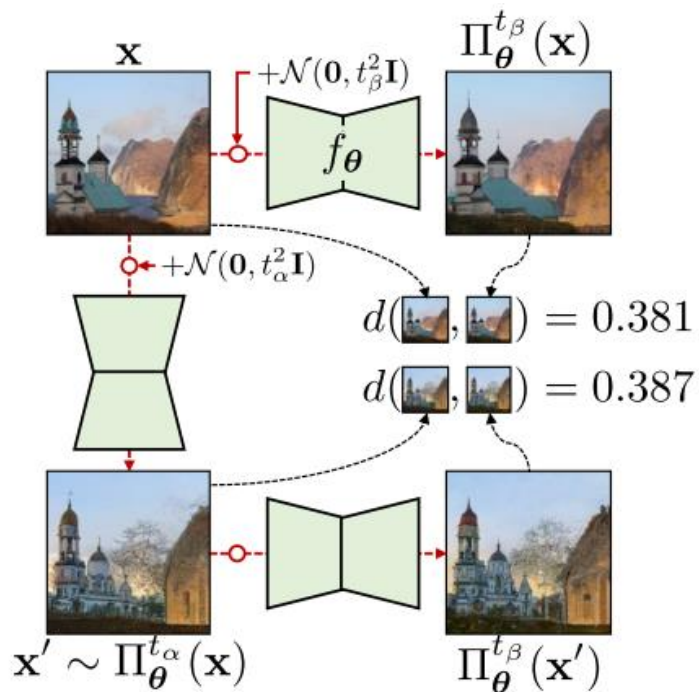


# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

방법:

- ① In-distribution 으로 학습된 consistency model 준비
- ② Background 정보의 의존하는 부분을 지워서 semantic difference 에 집중할 수 있게 하는 projection regret score 를 계산



$$S_{\text{PR}} = 0.381 - 0.387 = -0.006$$

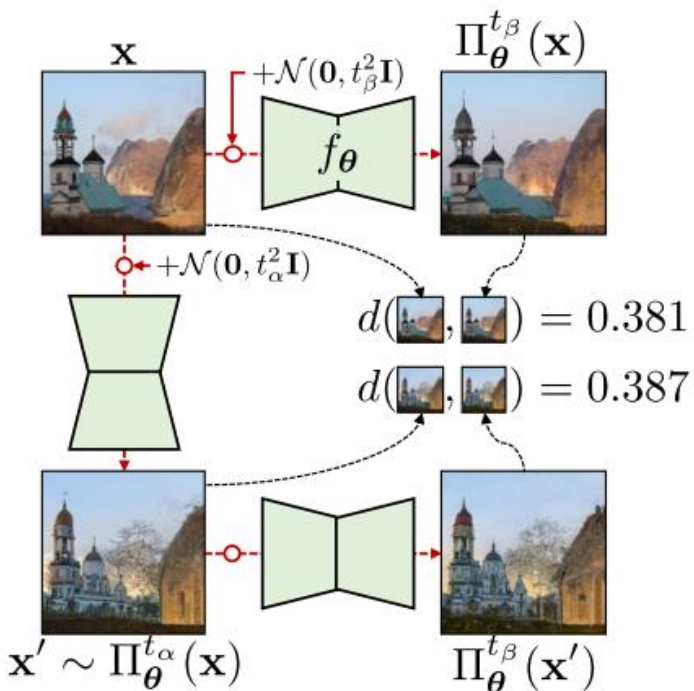
In-distribution

# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

방법:

- ① In-distribution 으로 학습된 consistency model 준비
- ② Background 정보의 의존하는 부분을 지워서 semantic difference 에 집중할 수 있게 하는 projection regret score 를 계산



$$S_{PR} = 0.381 - 0.387 = -0.006$$

In-distribution

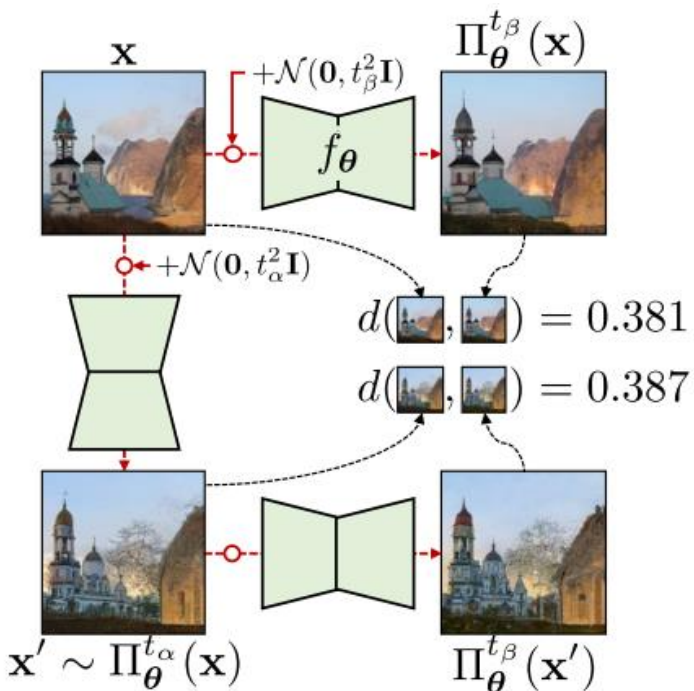
*Image Distance ( $d$ ) = semantic difference + background difference*  
LPIPS

# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

방법:

- ① In-distribution 으로 학습된 consistency model 준비
- ② Background 정보의 의존하는 부분을 지워서 semantic difference 에 집중할 수 있게 하는 projection regret score 를 계산



$$S_{PR} = 0.381 - 0.387 = -0.006$$

In-distribution

Image Distance ( $d$ ) = semantic difference + background difference  
LPIPS

$$d\left(x, \Pi_\theta^{t_\beta}(x)\right) = \text{semantic difference}(x, \Pi_\theta^{t_\beta}(x)) + \text{background difference}(x, \Pi_\theta^{t_\beta}(x))$$

$$d\left(x', \Pi_\theta^{t_\beta}(x')\right) = \text{semantic difference}(x', \Pi_\theta^{t_\beta}(x')) + \text{background difference}(x', \Pi_\theta^{t_\beta}(x'))$$

$\alpha$  는 background 가 유지되는 수준의 noise step

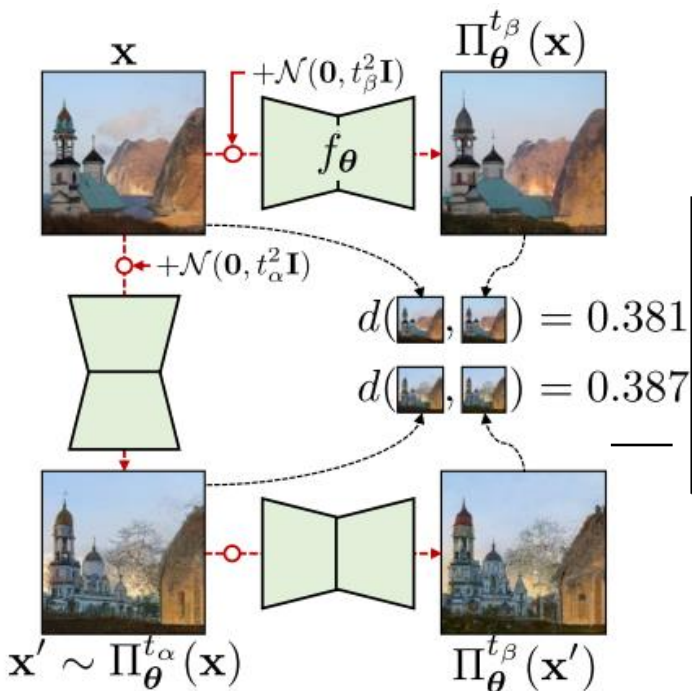
$\beta$  는  $\alpha$  보다는 크지만 background 정보가 유지되는 수준의 noise step

# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

방법:

- ① In-distribution 으로 학습된 consistency model 준비
- ② Background 정보의 의존하는 부분을 지워서 semantic difference 에 집중할 수 있게 하는 projection regret score 를 계산



$$S_{PR} = 0.381 - 0.387 = -0.006$$

In-distribution

Image Distance ( $d$ ) = semantic difference + background difference  
LPIPS

$$d\left(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x})\right) = \text{semantic difference}(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x})) + \text{background difference}(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x}))$$

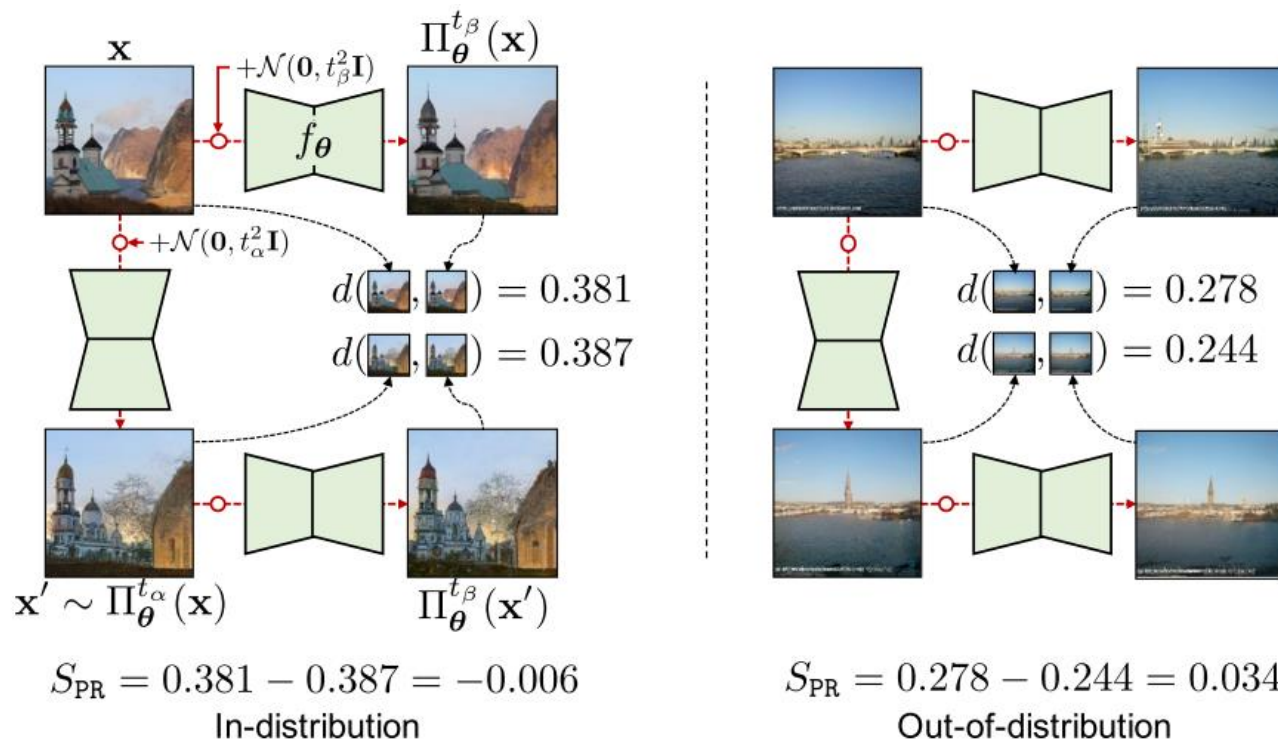
$$d\left(\mathbf{x}', \Pi_{\theta}^{t_{\beta}}(\mathbf{x}')\right) = \text{semantic difference}(\mathbf{x}', \Pi_{\theta}^{t_{\beta}}(\mathbf{x}')) + \text{background difference}(\mathbf{x}', \Pi_{\theta}^{t_{\beta}}(\mathbf{x}'))$$

$$\begin{aligned} \text{Projection Regret score} &= d\left(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x})\right) - d\left(\mathbf{x}', \Pi_{\theta}^{t_{\beta}}(\mathbf{x}')\right) = \\ &= \text{semantic difference}\left(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x})\right) - \text{semantic difference}(\mathbf{x}', \Pi_{\theta}^{t_{\beta}}(\mathbf{x}')) \end{aligned}$$

# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

방법:



Out-of-distribution의 semantic  
변화를 잘 잡아낼 수 있게 함.

$$\text{Projection Regret score} = \text{semantic difference}(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x})) - \text{semantic difference}(\mathbf{x}', \Pi_{\theta}^{t_{\beta}}(\mathbf{x}'))$$

$$\text{개인적인 생각: } \text{Projection Regret score} = \text{semantic difference}(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x})) - \text{semantic difference}(\mathbf{x}, \Pi_{\theta}^{t_{\beta}}(\mathbf{x}'))$$

Background 정보가 변경되진 않지만, semantic 정보가 변경되었을 경우를 out-of-distribution 으로 산출하고 싶은 것인데,

$\mathbf{x}'$  과정에서 이미 semantic 변화가 일어났을 수도 있으므로  $\mathbf{x}'$  보다는  $\mathbf{x}$ 와 비교하는 게 나을 것 같음

# Methods

- [NeurIPS 2023] **Projection Regret: Reducing Background Bias for Novelty Detection via Diffusion Models**

## 기여점:

- ① 기존 OOD detection 방법보다 우수한 성능을 보임
- ② Consistency models 을 적용함으로써 inference 속도 개선
- ③ Background 정보를 무시하는 새로운 OOD score 산출 방식 제안

Table 1: Out-of-distribution detection performance (AUROC) under various in-distribution vs out-of-distribution tasks. **Bold** and underline denotes the best and second best methods.

| Method                           | C10 vs       |              |              |              | C100 vs      |              |              | SVHN vs      |              |
|----------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                                  | SVHN         | CIFAR100     | LSUN         | ImageNet     | SVHN         | C10          | LSUN         | C10          | C100         |
| <i>Diffusion Models [26, 28]</i> |              |              |              |              |              |              |              |              |              |
| Input Likelihood [6]             | 0.180        | 0.520        | -            | -            | 0.193        | 0.495        | -            | 0.974        | 0.970        |
| Input Complexity [7]             | 0.870        | 0.568        | -            | -            | 0.792        | 0.468        | -            | 0.973        | 0.976        |
| Likelihood Regret [8]            | 0.904        | 0.546        | -            | -            | 0.896        | 0.484        | -            | 0.805        | 0.821        |
| MSMA [17]                        | <u>0.992</u> | 0.579        | 0.587        | 0.716        | 0.974        | 0.426        | 0.400        | 0.976        | 0.979        |
| LMD [18]                         | <u>0.992</u> | 0.607        | -            | -            | <b>0.985</b> | 0.568        | -            | 0.914        | 0.876        |
| <i>Consistency Models [21]</i>   |              |              |              |              |              |              |              |              |              |
| MSMA [17]                        | 0.707        | 0.570        | 0.605        | 0.578        | 0.643        | 0.506        | 0.559        | <u>0.985</u> | 0.981        |
| LMD [18]                         | 0.979        | <u>0.620</u> | <u>0.734</u> | <u>0.686</u> | 0.968        | <u>0.573</u> | <u>0.678</u> | 0.832        | 0.792        |
| Projection Regret (ours)         | <b>0.993</b> | <b>0.775</b> | <b>0.837</b> | <b>0.814</b> | 0.945        | <b>0.577</b> | <b>0.682</b> | <b>0.995</b> | <b>0.993</b> |

Table 5: AUROC of Projection Regret given different distance metric choices. **Bold** and underline denotes the best and second best methods.

| Method            | Distance $d$    | SVHN         | CIFAR100     | LSUN         | ImageNet     |
|-------------------|-----------------|--------------|--------------|--------------|--------------|
| LMD [18]          |                 | <u>0.979</u> | 0.620        | 0.734        | 0.686        |
| Projection Regret | SSIM [27]       | 0.328        | 0.629        | 0.650        | 0.620        |
|                   | LPIPS [20]      | <b>0.993</b> | <b>0.775</b> | <u>0.837</u> | <u>0.814</u> |
|                   | UNet (proposed) | 0.917        | <u>0.734</u> | <b>0.865</b> | <b>0.815</b> |



# Thank you!

**Seokho Moon**

`danny232@korea.ac.kr`

School of Industrial and Management Engineering, Korea University