ONE-CLASS CLASSIFICATION

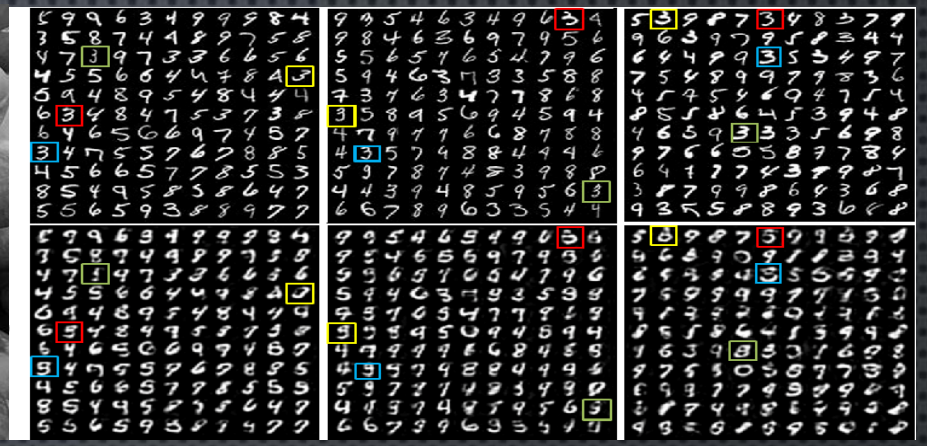
Mohammad Sabokrou

IPM Institute For Research In Fundamental Sciences

School of Computer Sciences

AI Research Group

Sabokro@ipm.ir



- Sudden dips or spikes in one or more sensor values that are unusual for “expected” operating conditions.
- Unusual images in a video feed, such as leaks, wildlife, or unexpected people.
- In almost any set of time series data, a sudden deviation that’s outside of the range you might expect given normal conditions.

https://www.arundo.com/thejourney/how-anomaly-detection-is-changing-industrial-operations

WHY SHOULD YOU CARE ABOUT ANOMALIES?

Accurate OCC == Prevent New Risks

1-Accurate : High Detection Rate, Acceptable False Positive

2-Fast : Enough fast to works as a real-world application

3-Safe

One class classification Problem:
In contrary with the conventional classifier only the information of one class (target class) is available.
Main Application: Anomaly detection

- Density Estimation
- Reconstruction Error

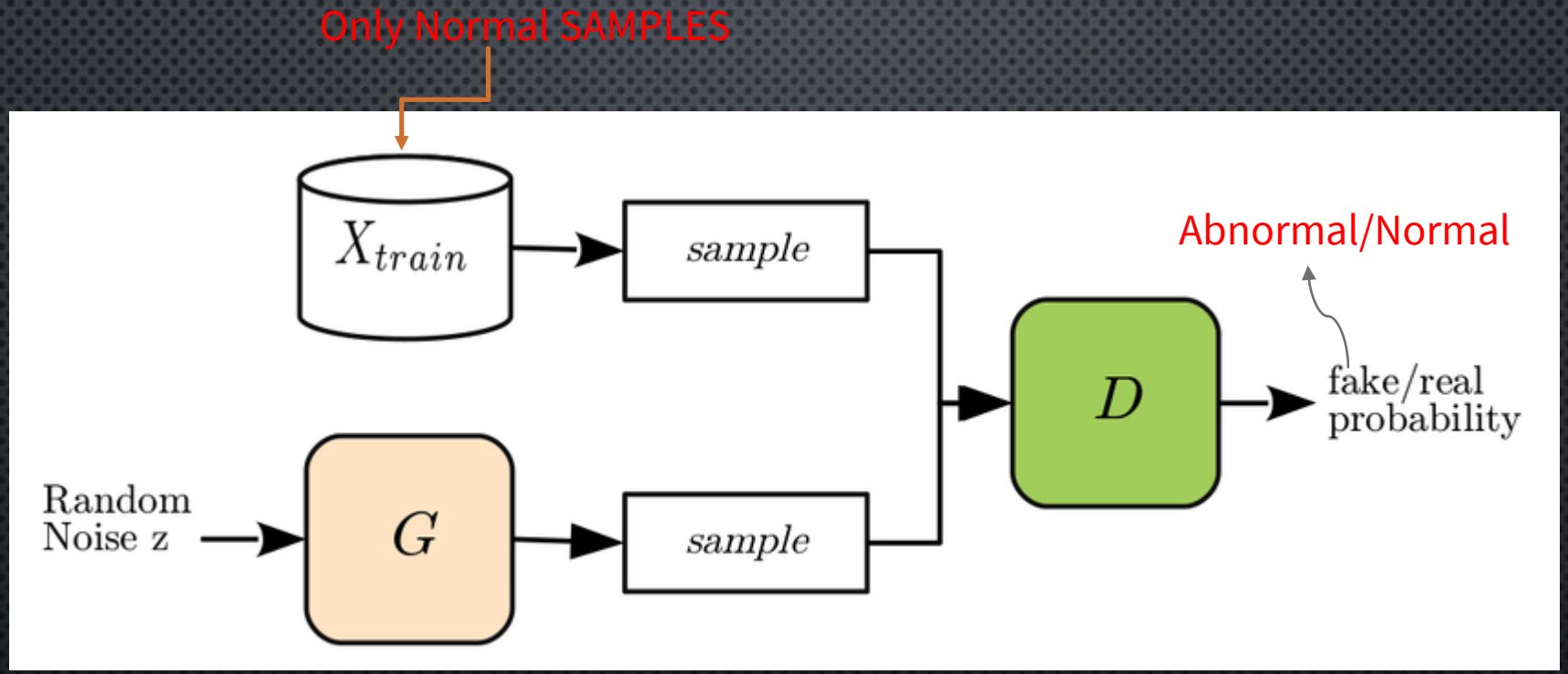
Deep Learning: Learning a deep network in absence of one networks is not feasible

- ❑ Learning the network by imposing a constraint on it: Minimizing Reconstruction error
- ❑ Simultaneously Generate the abnormal Samples: Generate with GAN

Train an Encoder-Decoder Network on only normal sample?

$$\mathcal{L}_{\mathcal{R}} = \|X - X'\|^2$$

GAN for OCC


Only Normal SAMPLES
X_{train}
sample
Abnormal/Normal
D
fake/real
probability
Random
Noise z
G
sample
Simultaneously Generated Abnormal Samples

Adversarially Learned One-Class Classifier for Novelty Detection

Computer Vision and Pattern Recognition (CVPR) 2018

Problem Statement (One-Class Classification)

Training

Testing

Applications:

- Novelty Detection
  - Outlier
  - Anomaly

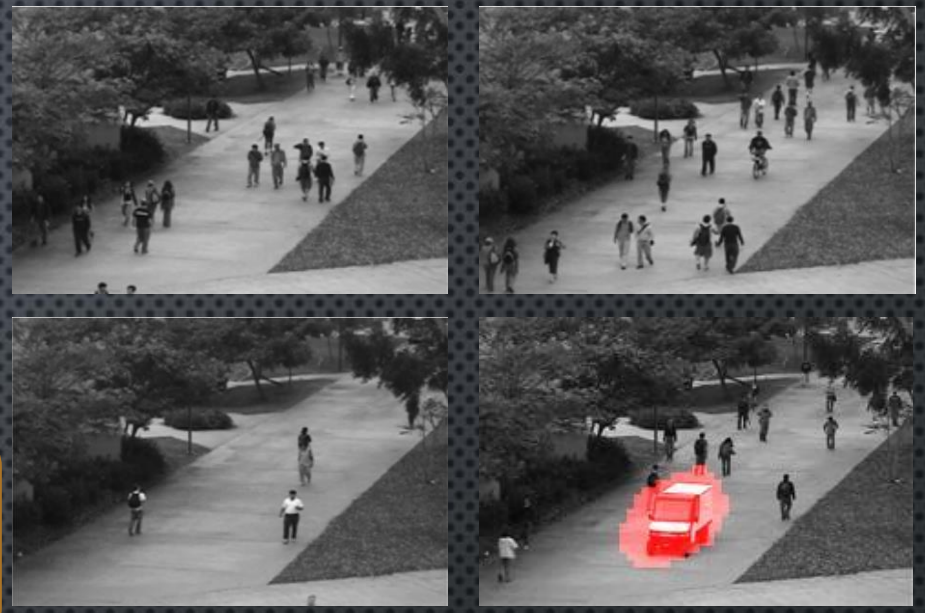
Challenges

- In reality, the novelty class is
  - absent during training,
  - poorly sampled, or
  - not well defined

No samples to train based on

Too few samples (highly imbalanced classification)

What is novelty?



- Due to the unavailability of data from the novelty class, training an end-to-end deep network is a cumbersome task.

Method

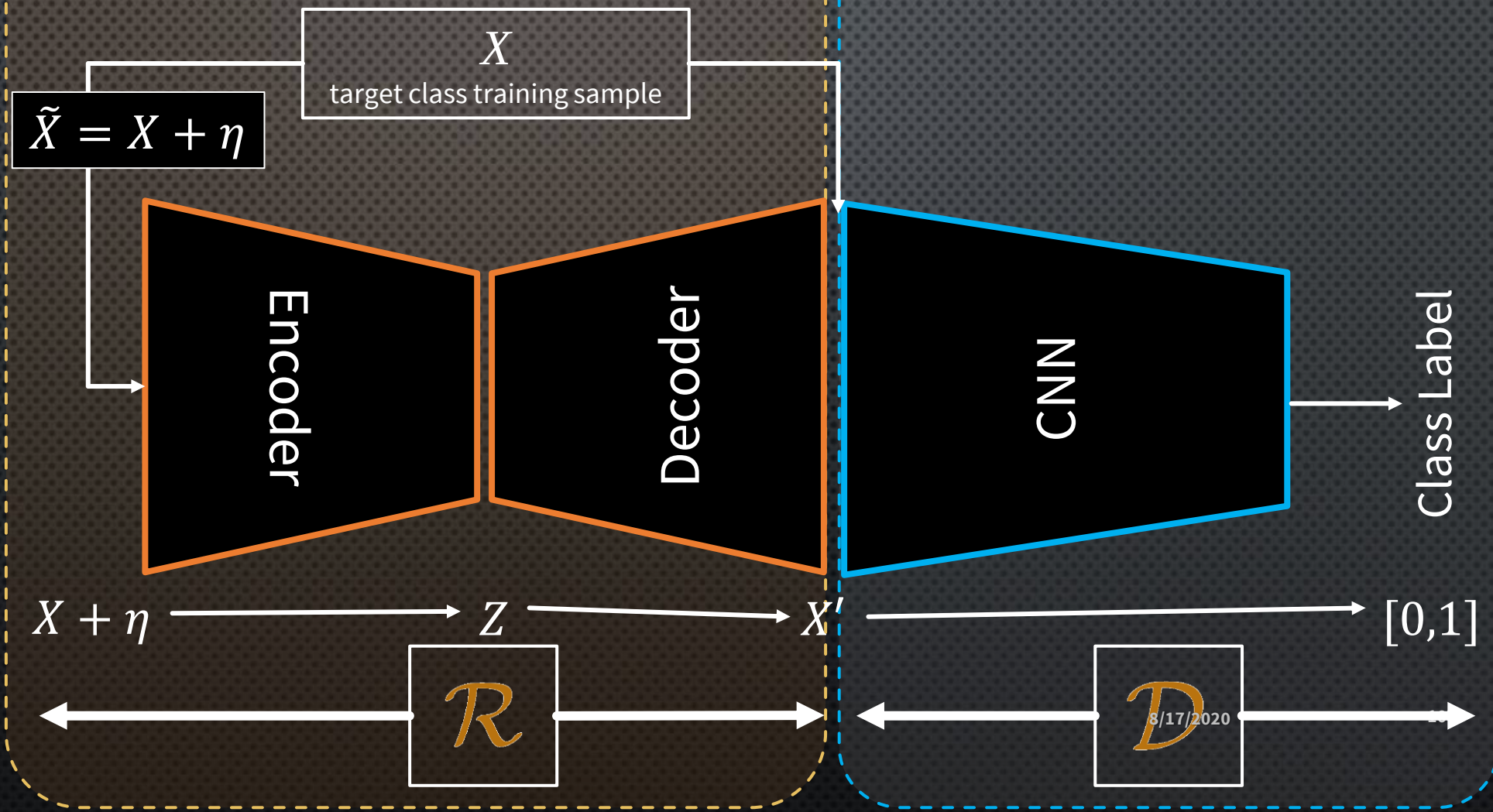
Learns to Reconstruct the positive samples (samples from target class)

Learns the concept characterized by the space of all positive samples (novelty Detector)

If exposed to negative samples (novelty), $\mathcal{R}$ decimates/distorts them.

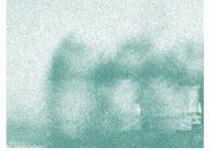
The two networks compete during training, but collaborate during testing.

Using the two networks at the testing time improves the results.



Method (cont'd)

Better discrimination power

	Noisy Inlier Samples		Outlier Samples	
X				
$\mathcal{R}(X)$				
$\mathcal{D}(X)$	0.75	0.72	0.53	0.27
$\mathcal{D}(\mathcal{R}(X))$	**0.85**	**0.91**	0.25	0.10

Joint Training of $\mathcal{R}$+$\mathcal{D}$

$\mathcal{R}$ ⟹ $\tilde{X} = (X \sim p_t) + (\eta \sim \mathcal{N}(0, \sigma^2 \mathbf{I})) \longrightarrow X' \sim p_t$

$\mathcal{D}$ ⟹ $\mathcal{R}(\tilde{X}) \sim p_t$? ✓ ✗

$$\mathcal{L}_{\mathcal{R}} = \|X - X'\|^2$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{R}+\mathcal{D}} + \lambda \mathcal{L}_{\mathcal{R}}$$

$$\min_{\mathcal{R}} \max_{\mathcal{D}} \Big(\mathbb{E}_{X \sim p_t} [\log(\mathcal{D}(X))] + \mathbb{E}_{\tilde{X} \sim p_t + \mathcal{N}_\sigma} [\log(1 - \mathcal{D}(\mathcal{R}(\tilde{X})))] \Big)$$

Joint Training (Summary)

Outlier or novelty sample

Similar to denoising autoencoders (but for a target concept)

$$\mathcal{R}(X \sim p_t + \eta) \longrightarrow X' \sim p_t$$

New concept? Does not know what to do, maps it to unknown distribution

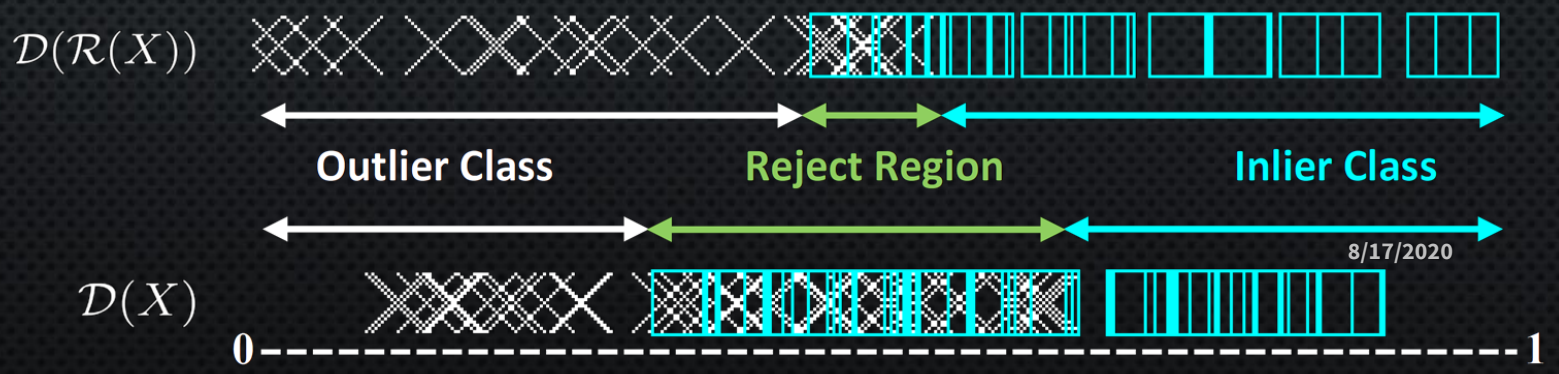
$$\mathcal{R}(\hat{X} \nsim p_t + \eta) \longrightarrow \hat{X}' \sim p_?$$

$\mathcal{D}$ is trained only to detect target samples, not novelty samples

$$\mathcal{D}(X' \sim p_t) > \mathcal{D}(\hat{X}' \nsim p_t)$$

Output of $\mathcal{R}$ is more separable than the original input images.

$$\mathcal{D}(\mathcal{R}(X \sim p_t)) - \mathcal{D}(\mathcal{R}(\hat{X} \nsim p_t)) > \mathcal{D}(X \sim p_t) - \mathcal{D}(\hat{X} \nsim p_t)$$

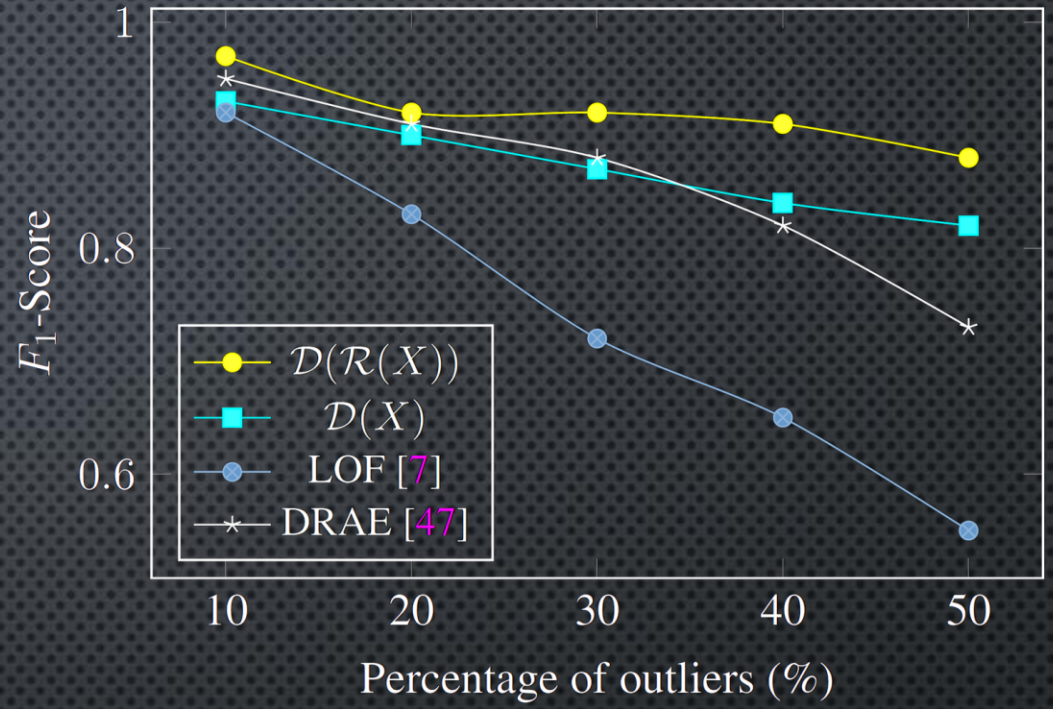


One-Class Classifier

$$\mathrm{OCC}_2(X) = \begin{cases} \text{Target Class} & \text{if } \mathcal{D}(\mathcal{R}(X)) > \tau, \\ \text{Novelty (Outlier)} & \text{otherwise.} \end{cases}$$

Experiments

- Outlier Detection (MNIST)
  - Trained to detect each digit separately
  - Other digits pose as outliers



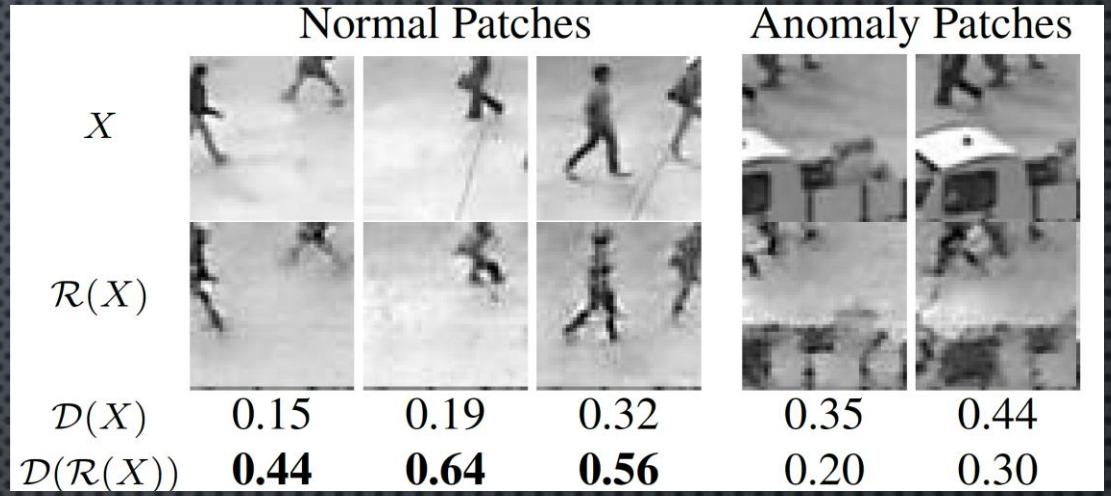
Experiments (cont'd)

- Outlier Detection (Caltech-256)
  - Similar to previous works [52], we repeat the procedure three times and use images from n={1; 3; 5} randomly chosen categories as inliers (i.e., target).
  - Outliers are randomly selected from the "clutter" category, such that each experiment has exactly 50% outliers.

		CoP [32]	REAPER [22]	OutlierPursuit [50]	LRR [24]	DPCP [45]	R-graph [52]	Ours $\mathcal{D}(X)$	Ours $\mathcal{D}(\mathcal{R}(X))$
1 outlier category	AUC	0.905	0.816	0.837	0.907	0.783	**0.948**	0.932	*0.942*
	F_1	0.880	0.808	0.823	0.893	0.785	0.914	*0.916*	**0.928**
3 outlier categories	AUC	0.676	0.796	0.788	0.479	0.798	0.929	*0.930*	**0.938**
	F_1	0.718	0.784	0.779	0.671	0.777	0.880	*0.902*	**0.913**
5 outlier categories	AUC	0.487	0.657	0.629	0.337	0.676	*0.913*	*0.913*	**0.923**
	F_1	0.672	0.716	0.711	0.667	0.715	0.858	*0.890*	**0.905**

Experiments (cont'd)

- Video Anomaly Detection (UCSD Ped2)



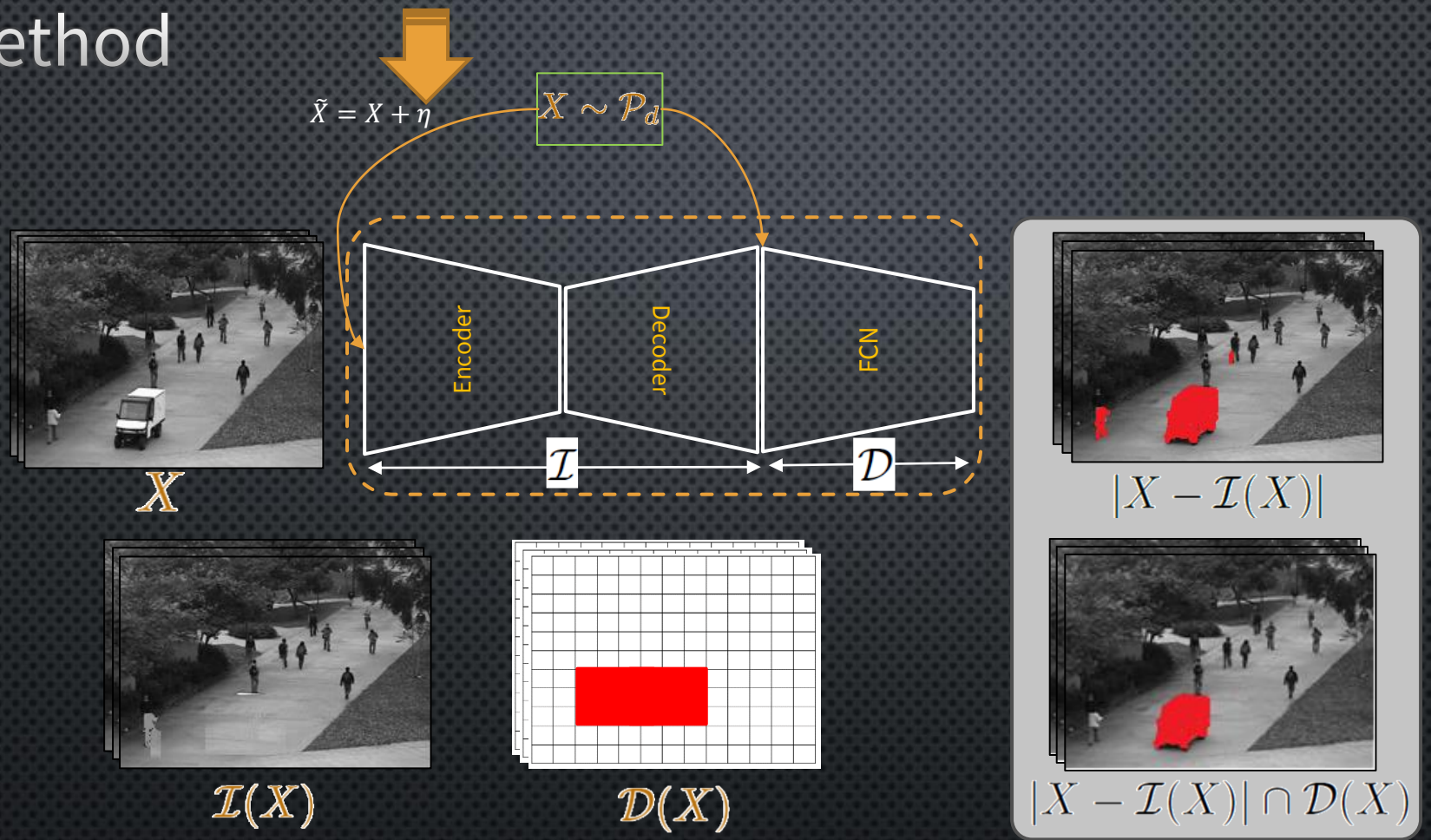
- Frame-level comparisons

Method	EER	Method	EER
IBC [6]	13%	RE [36]	15%
MPCCA [19]	30%	Ravanbakhsh *et al.* [34]	13%
MDT [26]	24%	Ravanbakhsh *et al.* [33]	14%
Bertini *et al.* [4]	30%	Dan Xu*et al.* [48]	17%
Dan Xu *et al.* [49]	20%	Sabokrou *et al.* [37]	19%
Li *et al.* [23]	18.5%	Deep-cascade [39]	**9%**
Ours - $\mathcal{D}(X)$	**16%**	Ours - $\mathcal{D}(\mathcal{R}(X))$	**13%**

AVID: Adversarial Visual Irregularity Detection

ACCV2019

Method



Pixel-level localization:

- high true positives
- Suffering from high false positive errors.

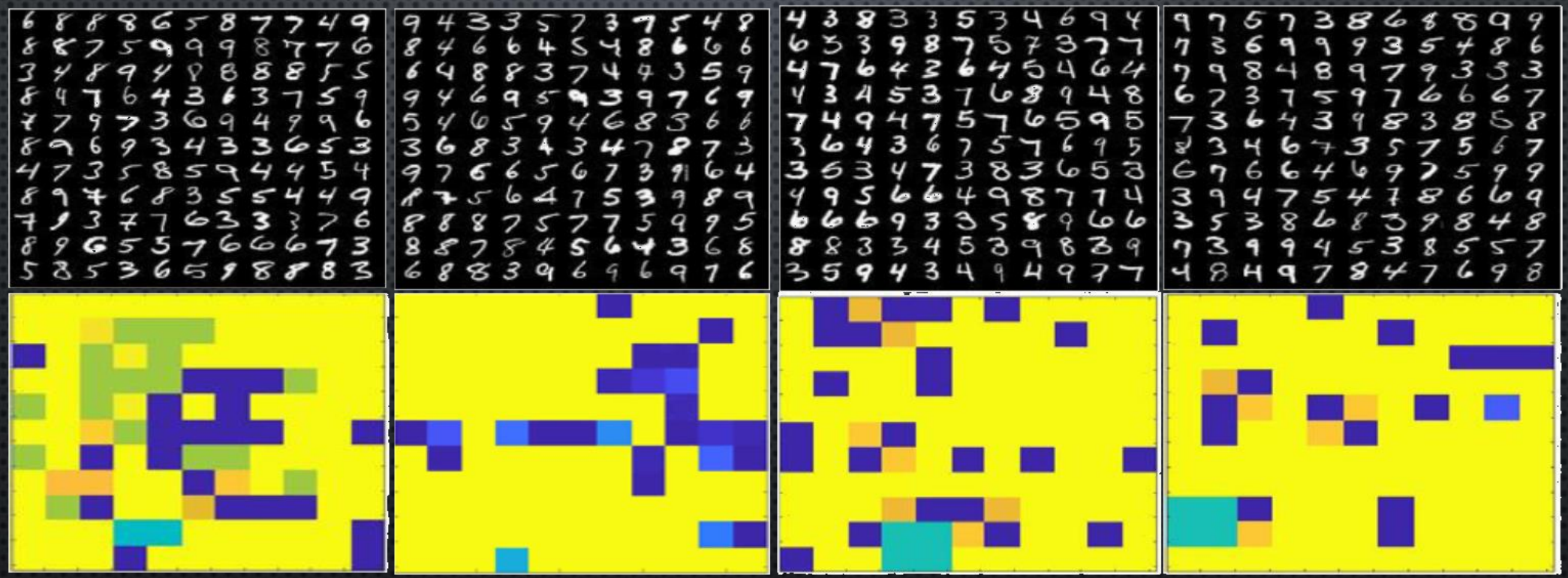
Patch-level localization:

- Overcome the problem of high false positive error
- Sacrificing the detection rate.

Results

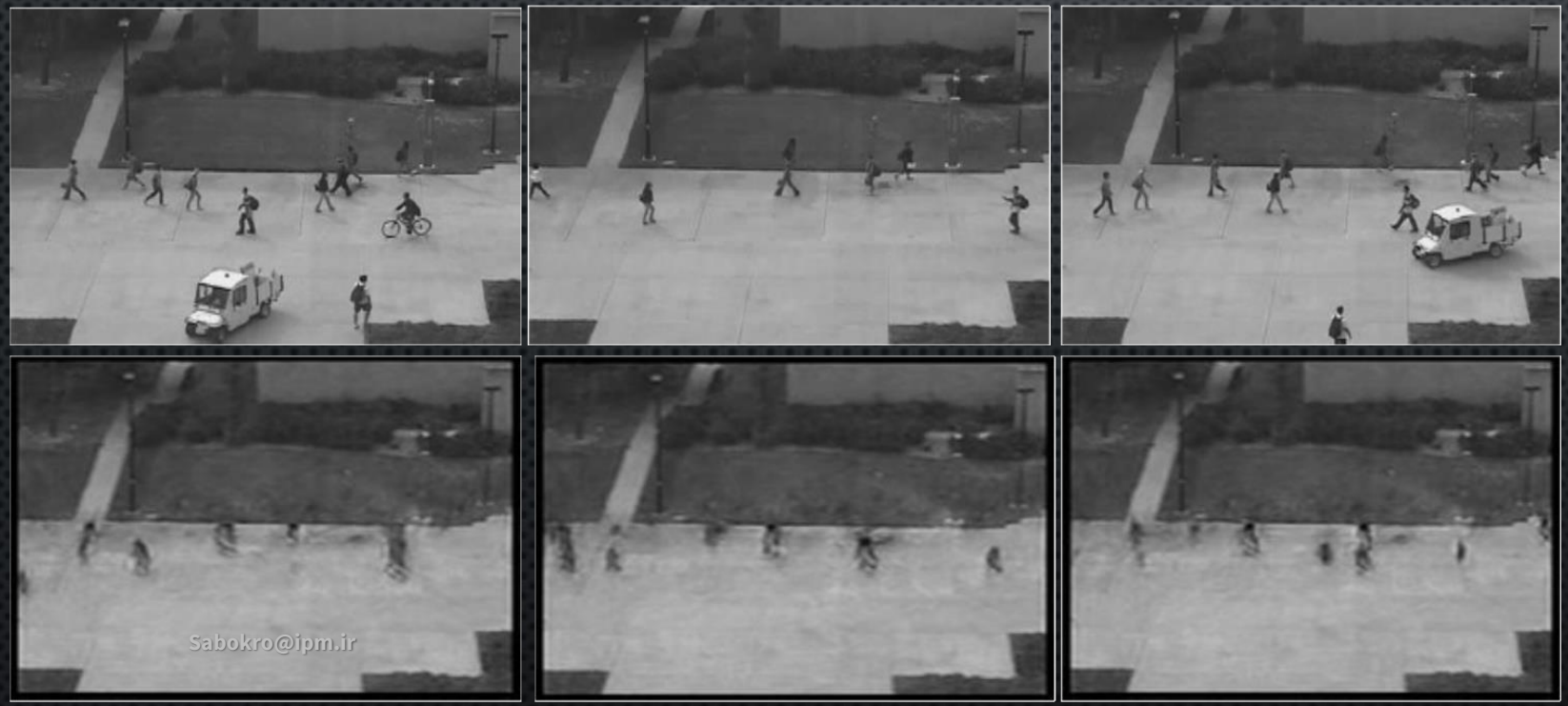
IR-MNIST

Results (cont'd)



Results (cont’d)

- UCSD Ped2 (Inpainted Images)



Results (cont'd)

- Comparisons on UCSD Ped2 (Frame-Level & Pixel-Level)

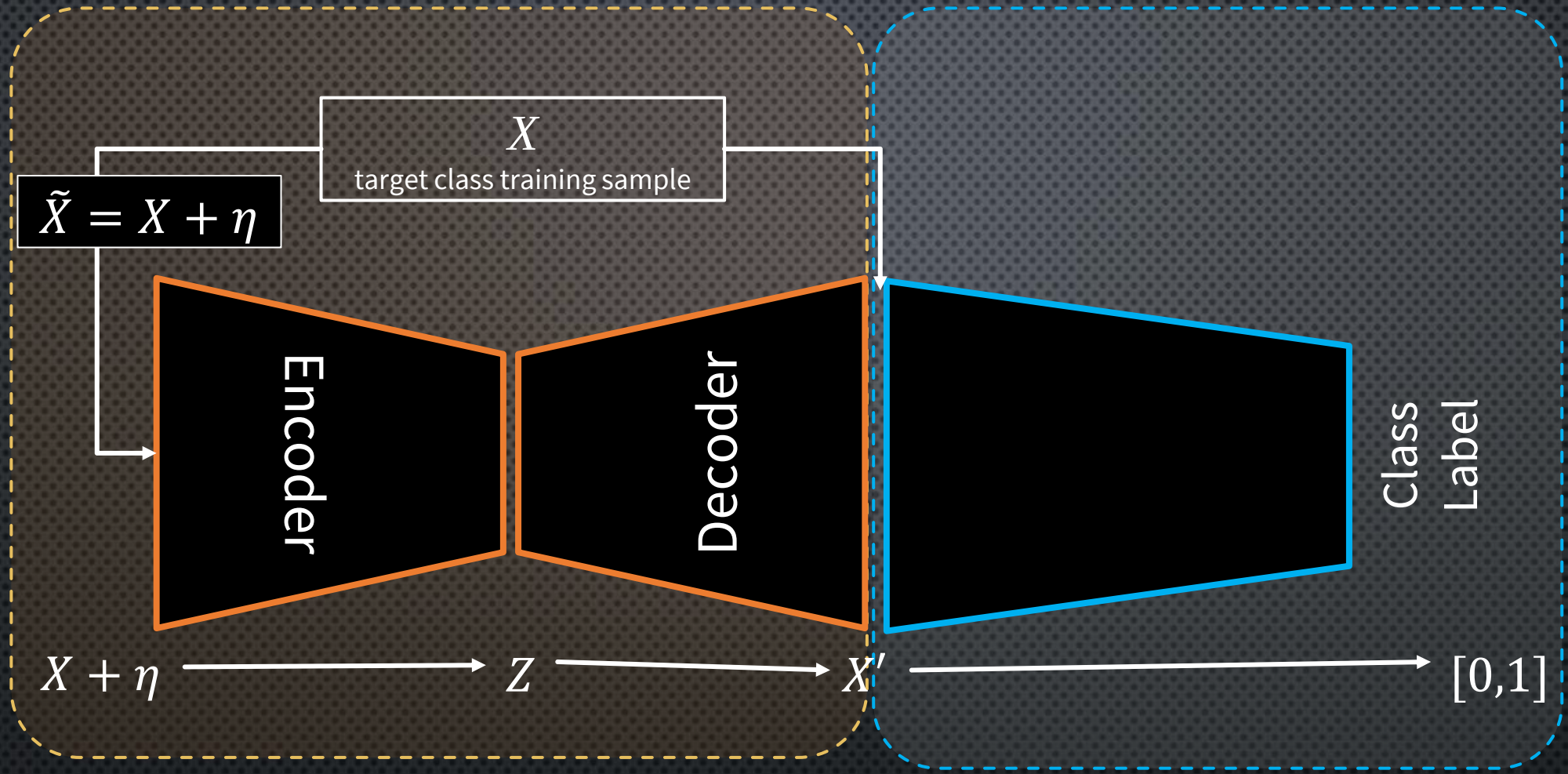
	IBC [1]	MPCCA [24]	MDT [2]	[3]	[26]	[44]	[7]	[26]	[27]	[9]	Ours (AVID)
FL	13	30	24	30	20	18.5	15	17	19	**9**	14
PL	26	71	54	–	42	29.5	–	42	30	24	**15.6**

- Comparisons on UMN Dataset

	Chaotic invariant [45]	SF [2]	Cong *et al.* [31]	Saligrama *et al.* [46]	Li *et al.* [44]	Ours (AVID)
EER	5.3	12.6	2.8	3.4	3.7	**2.6**
AUC	99.4	94.9	**99.6**	99.5	99.5	**99.6**

G2D: Generate to Detect Anomaly

WACV2020



When is the appropriate time for stopping the learning process?

- Previous researches solve this problem as a One Class Classification (OCC) task where they train a reference model on all of the available samples.
- Generative Adversarial Networks (GANs) have achieved the most promising results for OCC.
  - Cumbersome and computationally expensive procedure

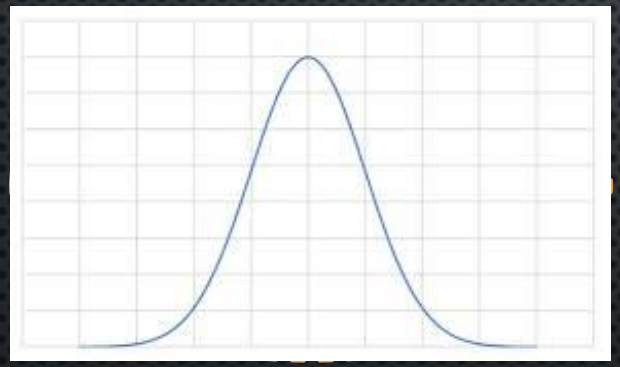
Finding the appropriate time for stopping the learning process to achieve the best performance without any validation samples of outlier class is very challenging and needs trial and error.

Highly dependent on hyper-parameters such as number of layers, kernels and the learning rate

Can We reduce the One-Class classification to a binary classification task?

Anomaly:
Random Samples

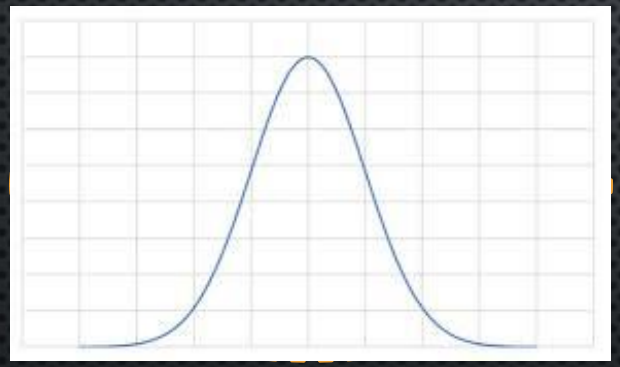
Class
Label

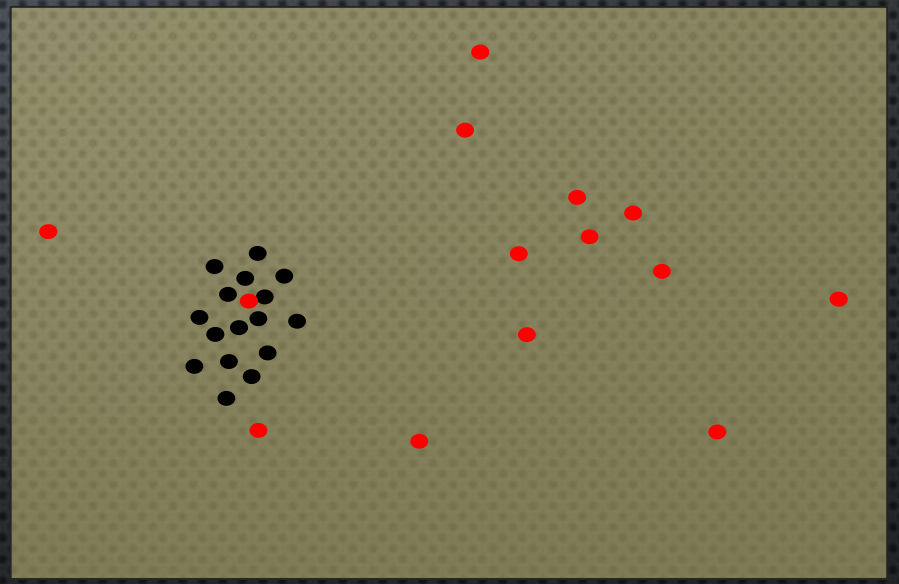


Class Label

Assuming that the representation of abnormal samples follow Gaussian distribution

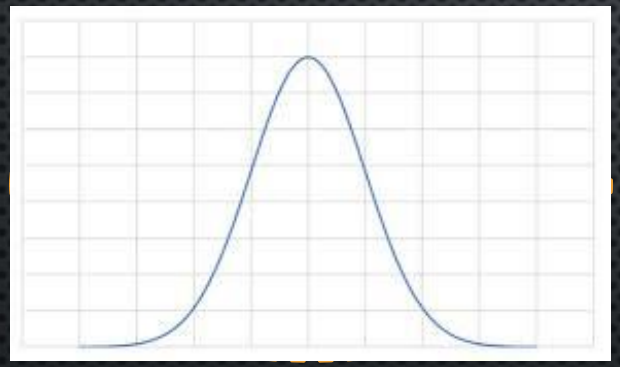
What is wrong with this idea?





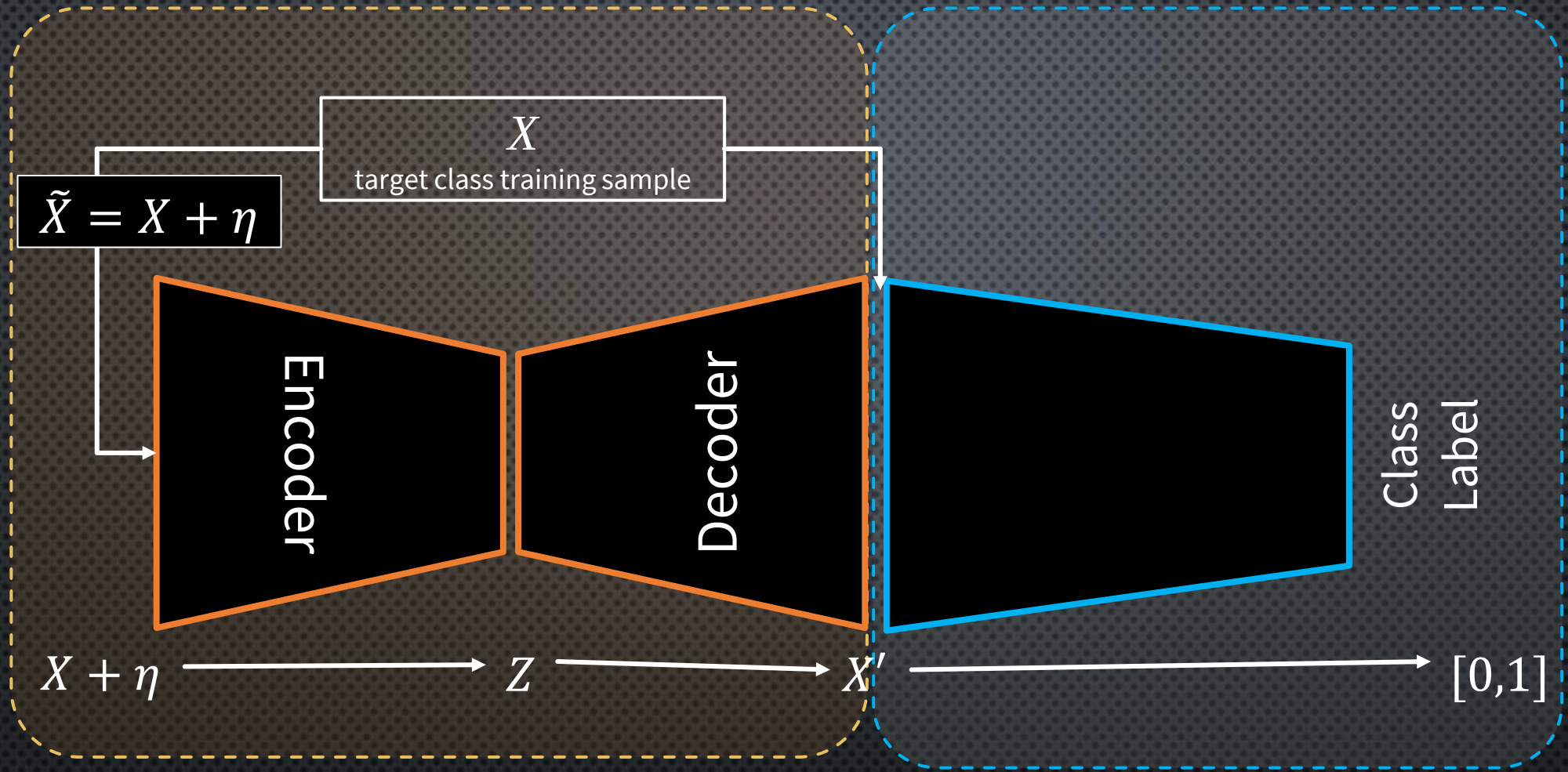
Assuming that the representation of abnormal samples follow Gaussian distribution

What is wrong with this idea?

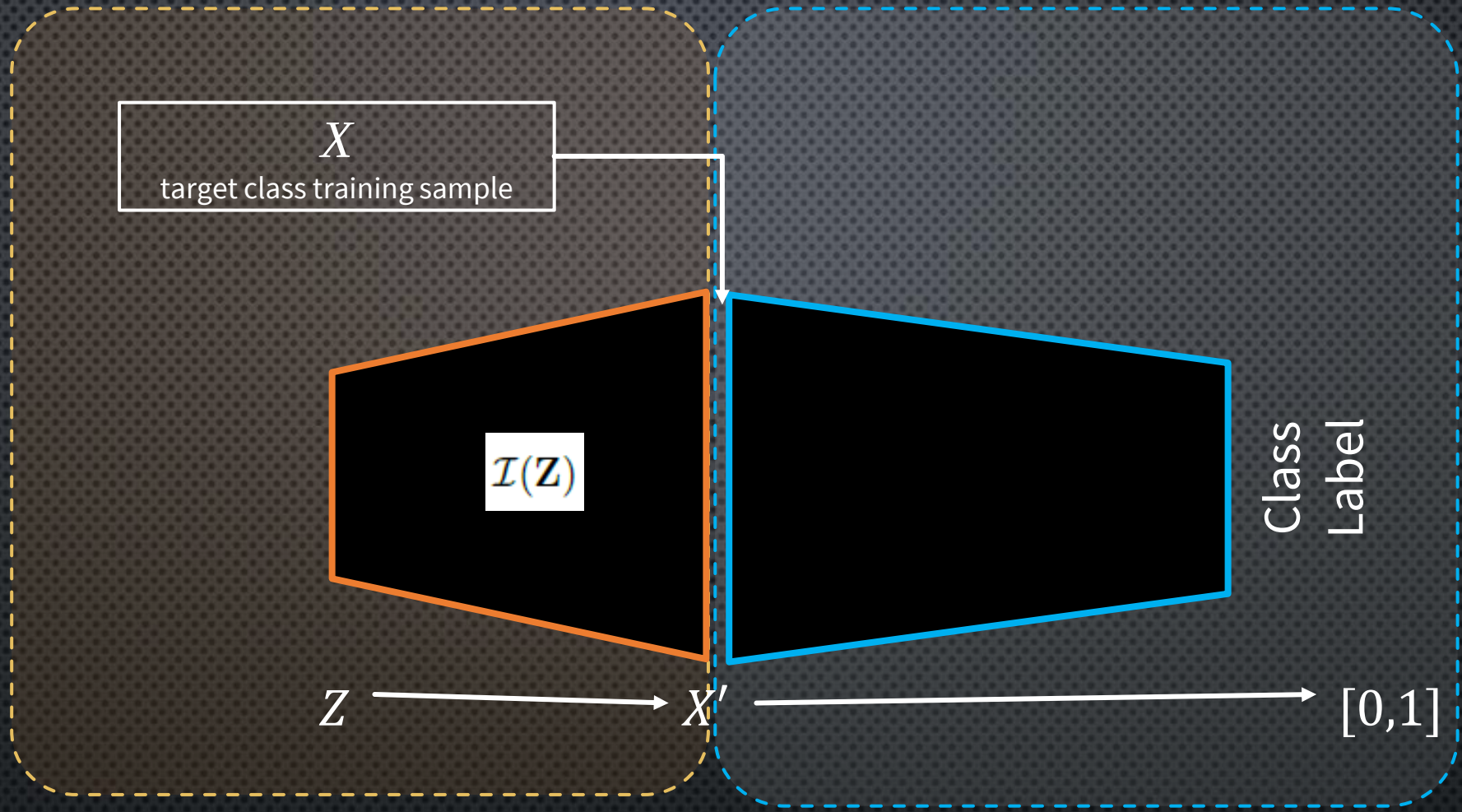


Assuming that the representation of abnormal samples follow Gaussian distribution

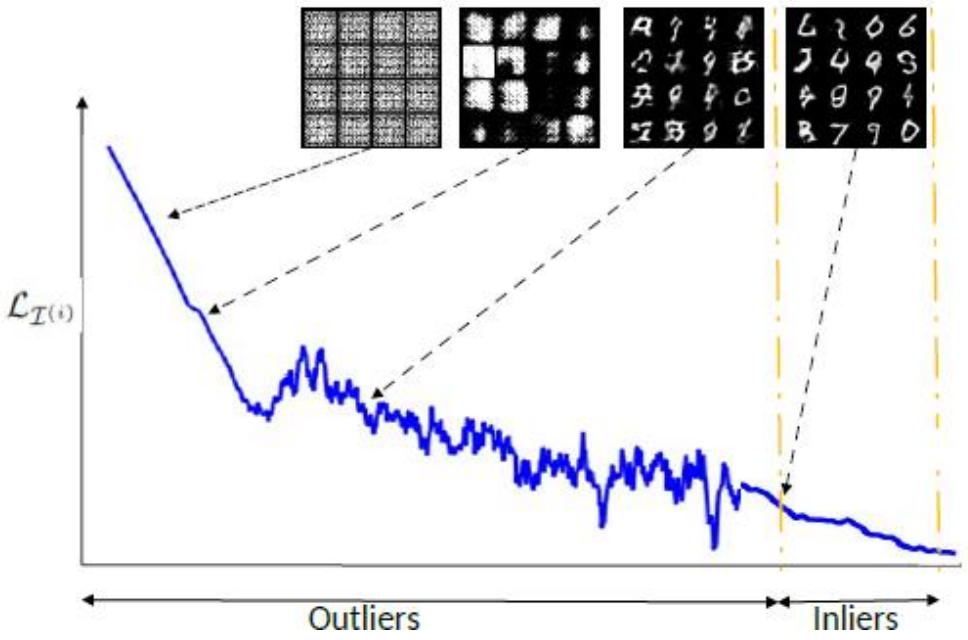
Idea: Generate the Abnormal samples which are semantically meaningful



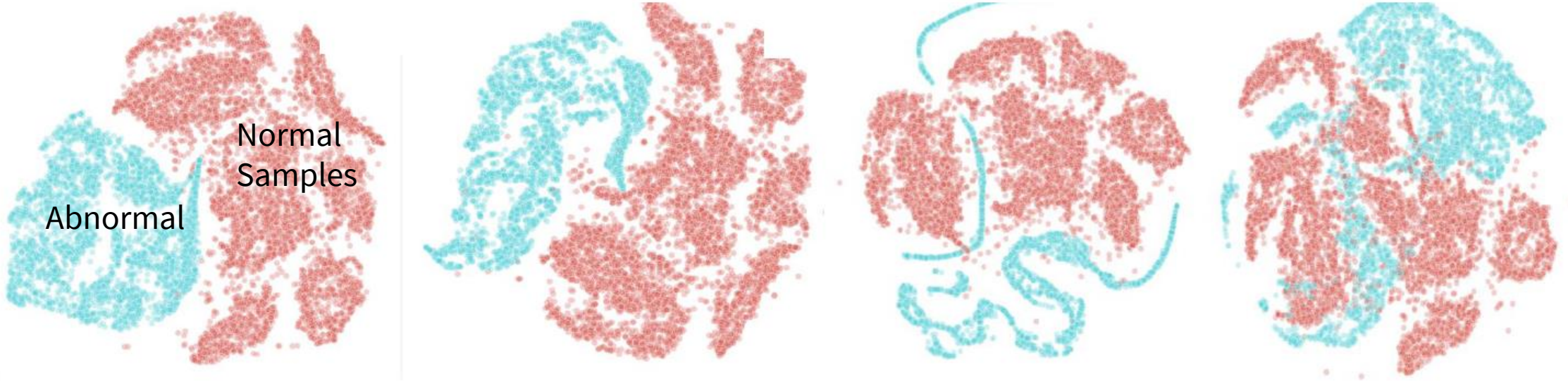
When is the appropriate time for stopping the learning process?



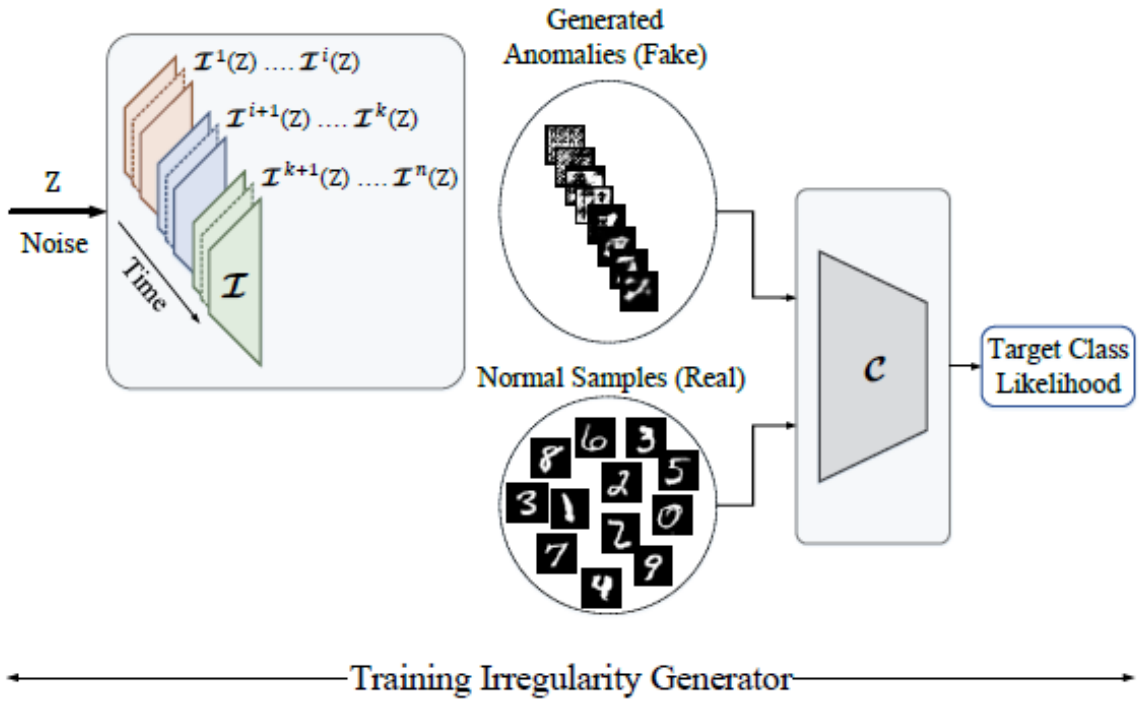
When is the appropriate time for stopping the learning process?



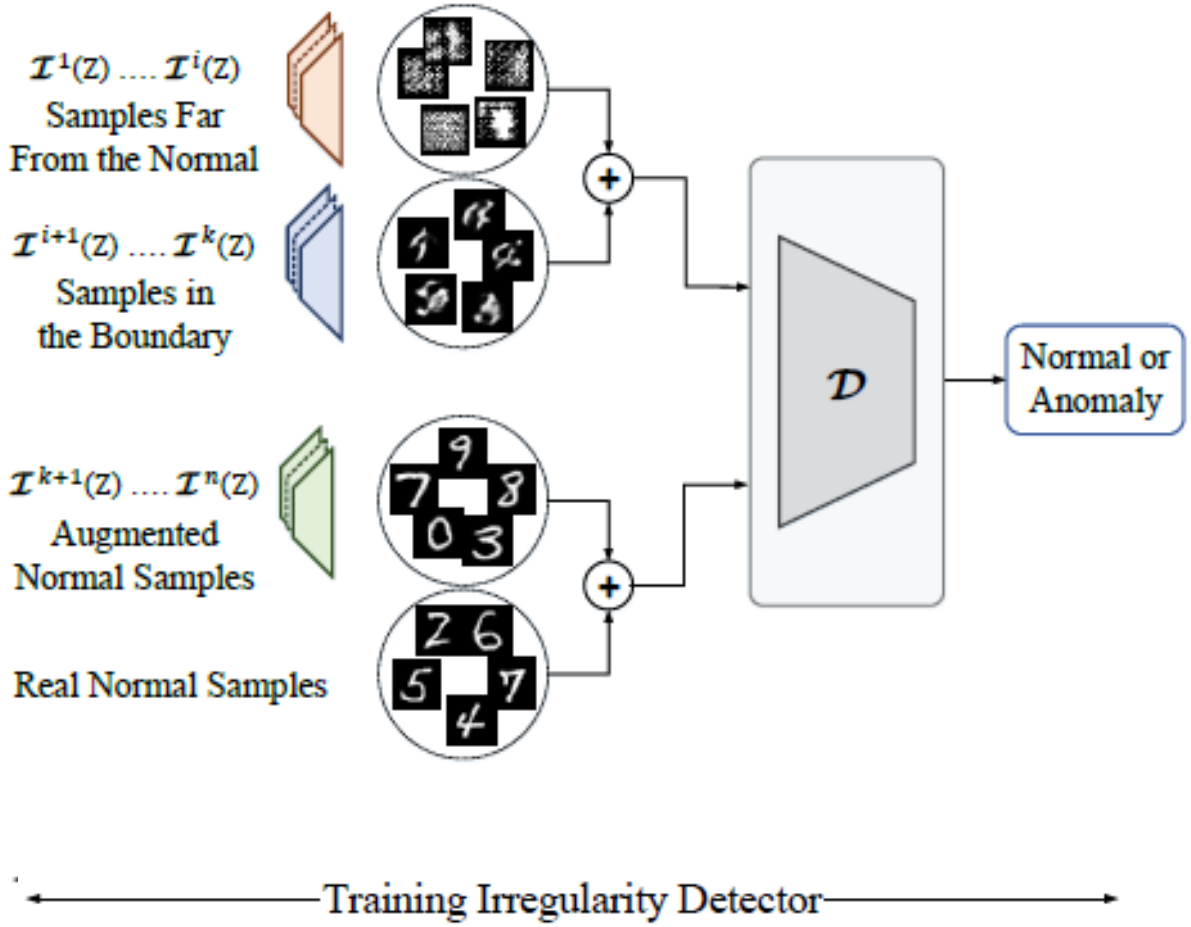
$$\mathcal{I}(\mathbf{Z}) = \begin{cases} \text{Random irregularity (}i.e.\text{, noise)}, & \text{if } L < \epsilon_1. \\ \text{Irregularity close to the boundary}, & \text{if } L < \epsilon_2. \\ \text{An inlier}, & \text{if } L < \epsilon_3. \end{cases}$$



Epochs



Two networks are jointly and adversarially trained on normal class data. In training duration, several models (with different weights) are considered as the irregularity generator.


$\mathcal{I}^1(z) \ldots \mathcal{I}^i(z)$
Samples Far From the Normal
$\mathcal{I}^{i+1}(z) \ldots \mathcal{I}^k(z)$
Samples in the Boundary
$\mathcal{I}^{k+1}(z) \ldots \mathcal{I}^n(z)$
Augmented Normal Samples
Real Normal Samples
$\mathcal{D}$
Normal or Anomaly
Training Irregularity Detector

OCC==BCC

Some Discussion!

- Anomaly Generation
- Anomaly Prediction
- XAI & Anomaly Detection
- Knowledge Leakage & Anomaly Detection
- Adverserail Examples

Team




Hiring

IPM School of Computer Science is seeking applicants with a strong track-record of publication (preferably top-tier conference publication) in all areas of computer science with an emphasis on Artificial Intelligence and Computer Security.

Thank You!

Any questions