Adversarially Learned One-Class Classifier for Novelty Detection

M. Sabokrou[1], M. Khalooei[2], M. Fathy[1], **Ehsan Adeli**[3]

[1] Institute for Research in Fundamental Sciences [2] Amirkabir University of Technology [3] Stanford University

CVPR 2018 SALT LAKE CITY • JUNE 18-22

Motivation and Problem Statement

Training

Testing

One-Class Classifier Applications:

- Novelty Detection
  - Outlier
  - Anomaly

- 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?

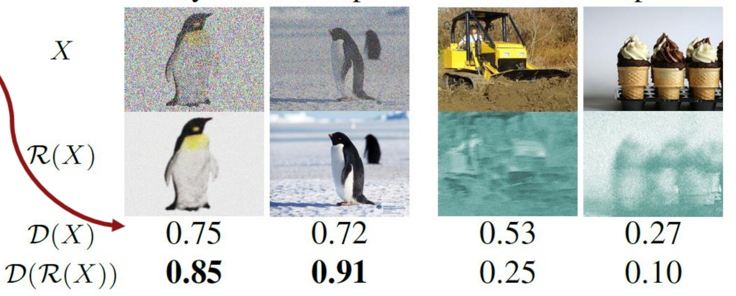
- Due to the unavailability of data from the novelty class, training an end-to-end deep network is challenging.

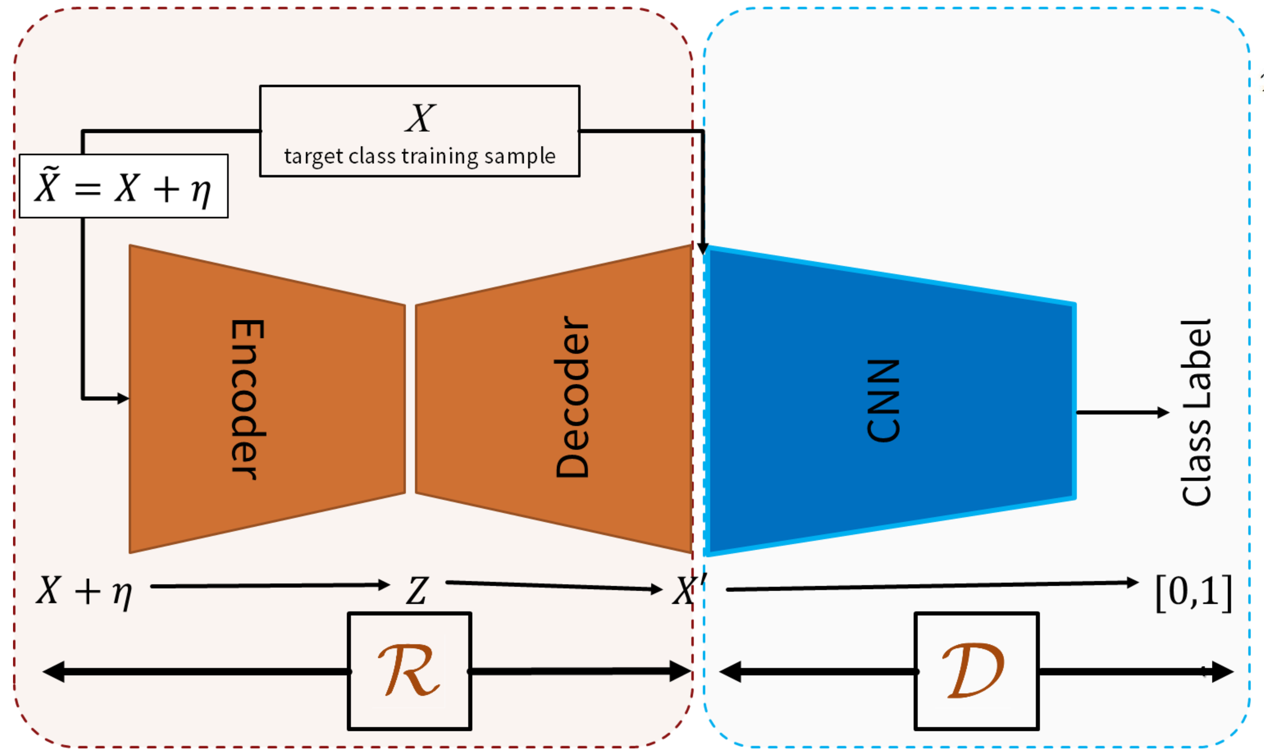
Method

Better discrimination power

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

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





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

The networks compete during training, but collaborate in testing.

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

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

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

$$\mathcal{D} \Rightarrow \mathcal{R}(\tilde{X}) \sim p_t \,? \checkmark \times$$

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

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

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

Similar to Generative Adversarial Networks (GANs)

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

Outlier or novelty sample

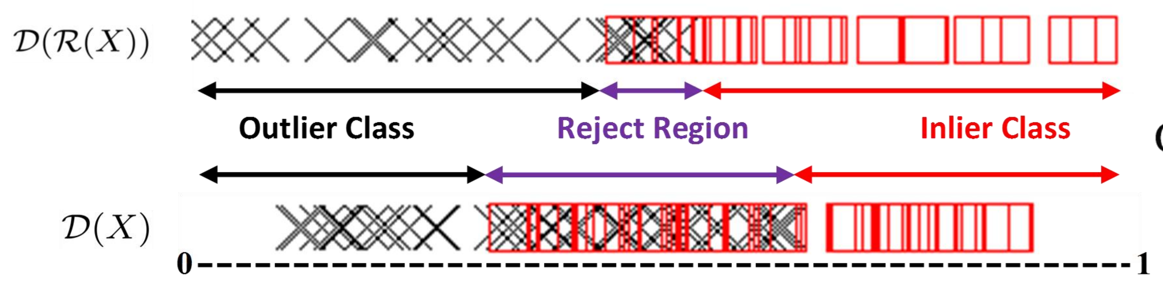
$$\mathcal{R}(\hat{X} \nsim p_t + \eta) \longrightarrow \hat{X}' \sim p_?$$

$\mathcal{D}$ 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.

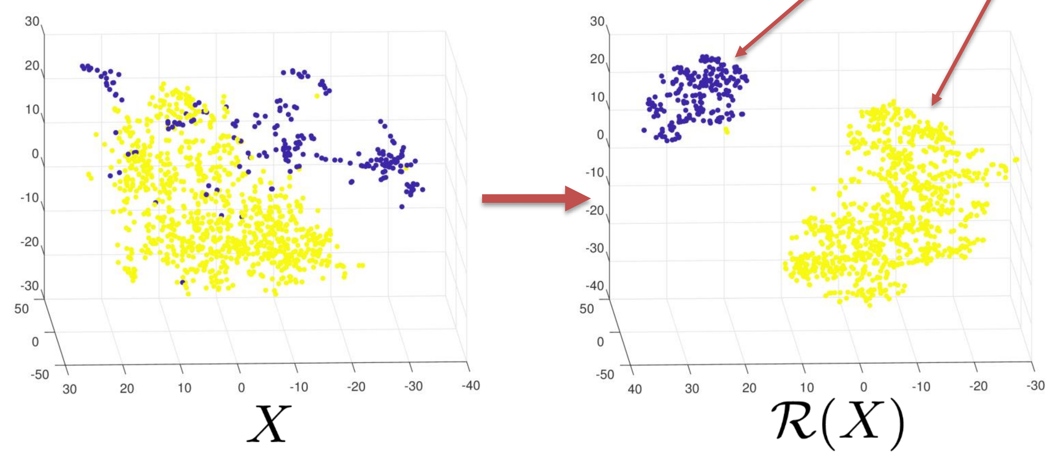
$$\mathcal{D}(\mathcal{R}(X \sim p_t)) - \mathcal{D}(\mathcal{R}(\hat{X} \nsim p_t)) > \mathcal{D}(X \sim p_t) - \mathcal{D}(\hat{X} \nsim p_t)$$

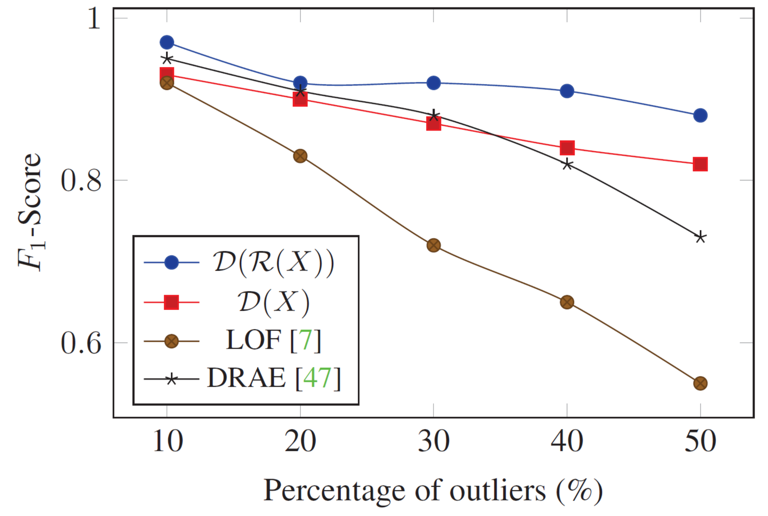


$$OCC_2(X) = \begin{cases} \text{Target Class} & \text{if } \mathcal{D}(\mathcal{R}(X)) > \tau, \\ \text{Novelty (Outlier)} & \text{otherwise.} \end{cases}$$

Experiments

- Outlier Detection (MNIST)
  - Trained to detect each digit separately
  - Other digits pose as outliers (10 to 50 percent)
  - Averaged results over all 10 digits are reported.





Trained with digit '1' as the target class

First row (X)
Second row $\mathcal{R}(X)$

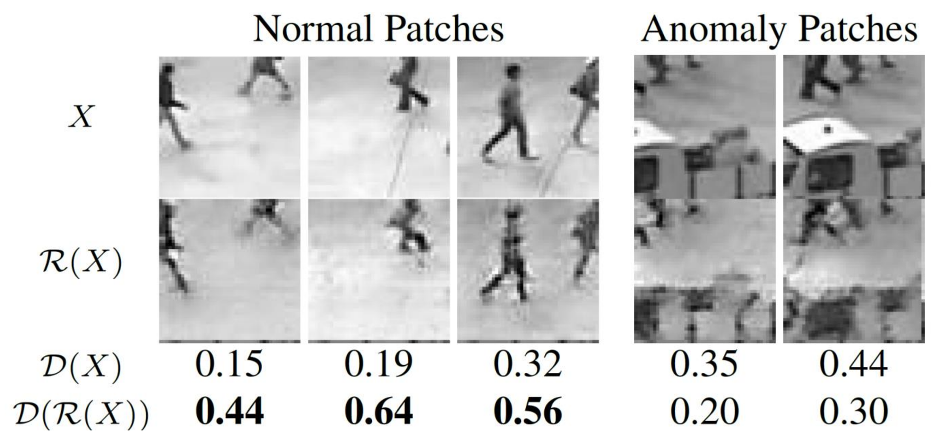
Reconstructs '1' properly, distorts others

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.

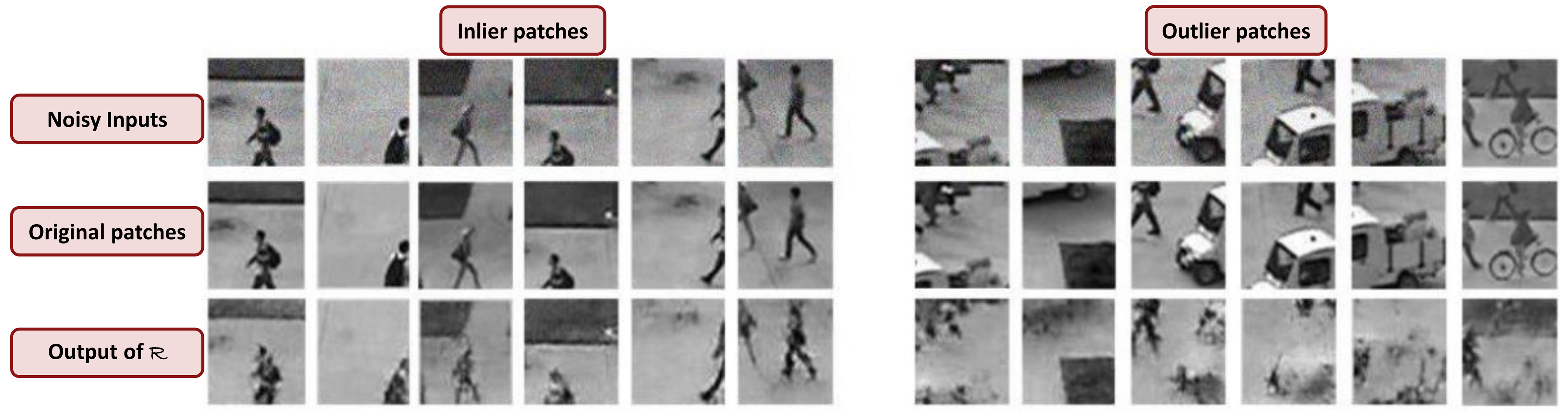
		CoP [32]	REAPER [22]	OutlierPursuit [50]	LRR [24]	DPCP [45]	R-graph [52]	Ours $\mathcal{D}(X)$	Ours $\mathcal{D}(\mathcal{R}(X))$
1 outlier category	AUC	0.905	0.816	0.837	0.907	0.783	**0.948**	0.932	*0.942*
	F_1	0.880	0.808	0.823	0.893	0.785	0.914	*0.916*	**0.928**
3 outlier categories	AUC	0.676	0.796	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**

- 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%**



Conclusion

- Unlike majority of GAN applications, here, both trained networks are used in testing.
- After training the model, $\mathcal{R}$ can reconstruct target class samples correctly, while it distorts samples that do not have the concept shared among the target class samples, which indeed helps $\mathcal{D}$.
- No significant problems with Mode Collapse, as $\mathcal{R}$ directly sees all possible samples of the target class data and implicitly learns the manifold spanned by the target data distribution.
- **Questions:** sabokro@ipm.ir, eadeli@cs.stanford.edu