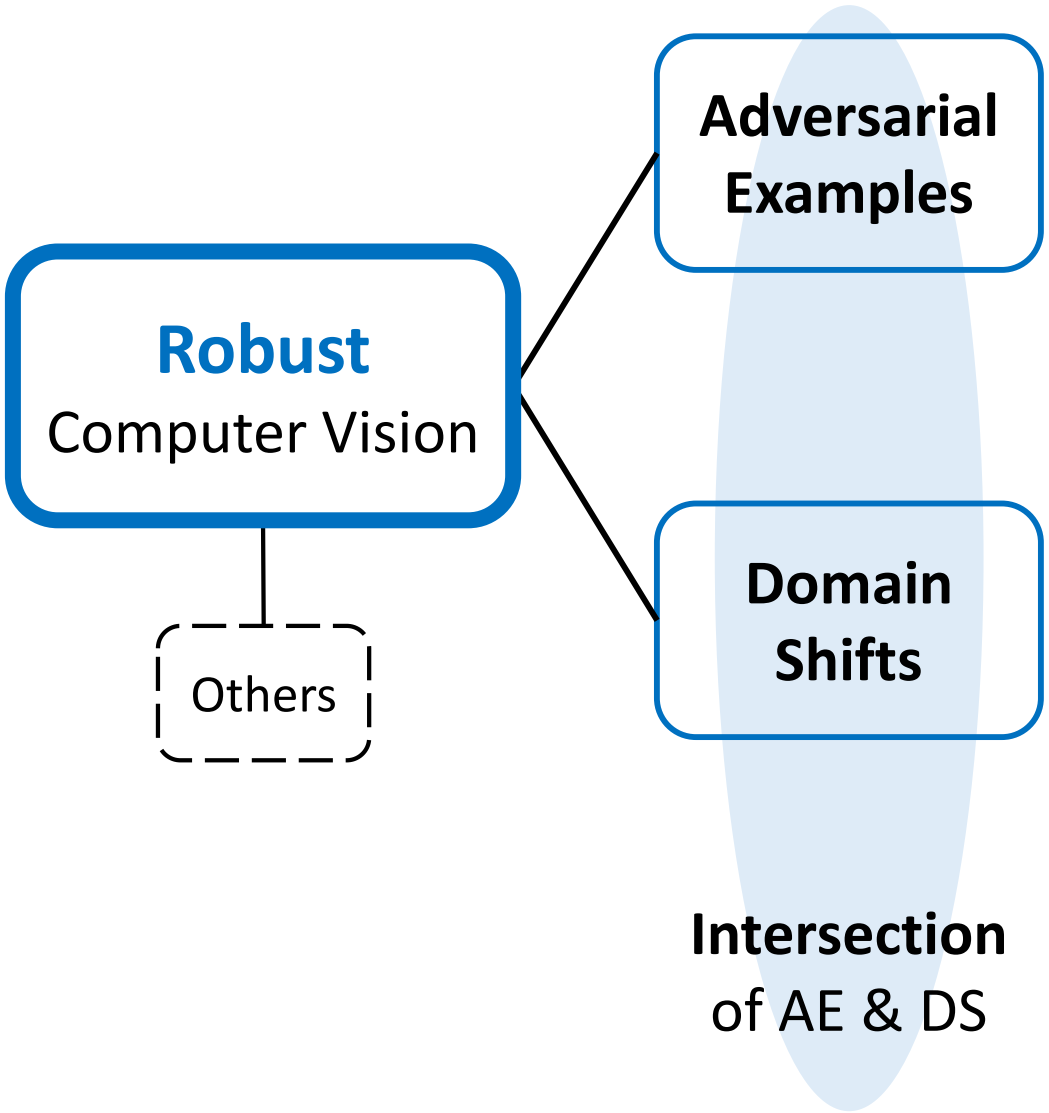
JOHNS HOPKINS
UNIVERSITY

Robust Computer Vision Against Adversarial Examples and Domain Shifts

Shao-Yuan Lo

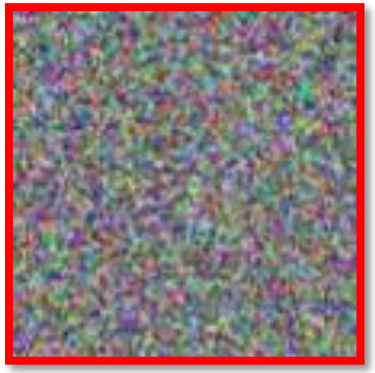
April 14, 2023


Robust
Computer Vision
Others
Adversarial Examples
Domain Shifts
Intersection
of AE & DS

Recall: Adversarial Examples

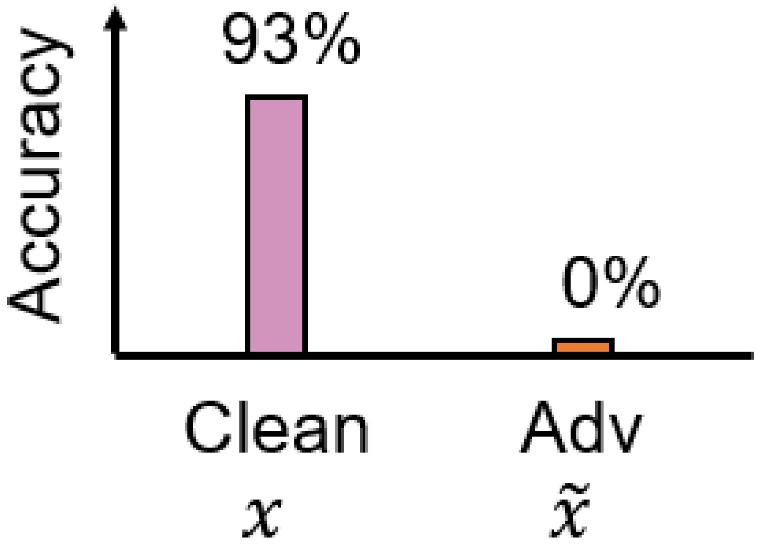
- Deep networks are **vulnerable** to adversarial examples.

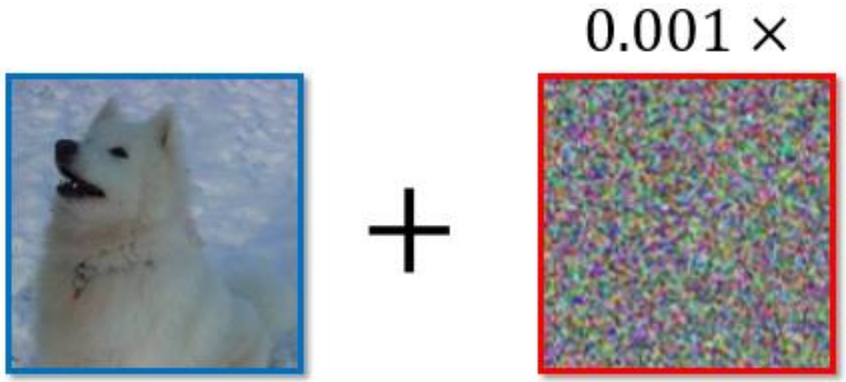
$f_\theta($  $) = "Dog"$

$f_\theta($  $+$ $0.001 \times$  $) = "Cat"$

Recall: Adversarial Examples

- Dataset: CIFAR-10
- Network: ResNet-50

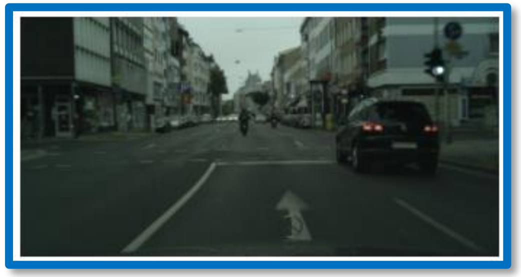


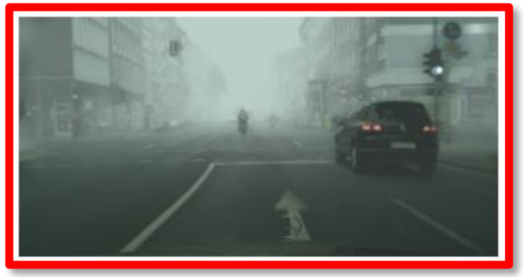


Recall: Domain Shifts

- **Scenario:** **Training (source) data** and **test (target) data** are from different domains (i.e. datasets).
- **Setting:** Given a **labeled source** dataset and an **unlabeled target** dataset, learn a model for the **target** domain.

Cityscapes





Foggy Cityscapes

Source domain

Target domain

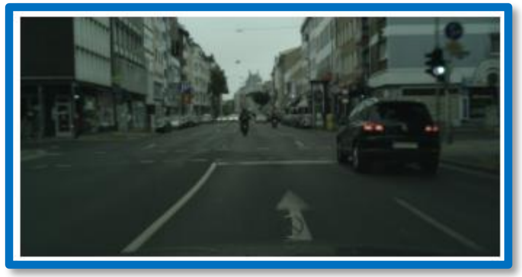
Virtual KITTI

KITTI

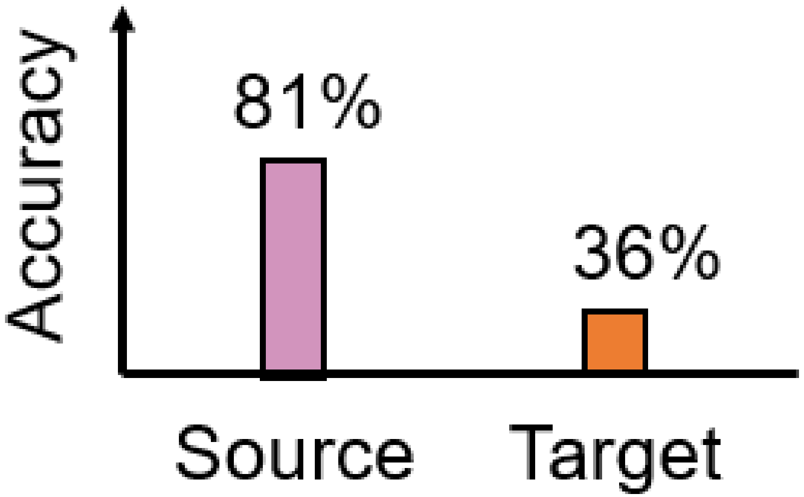
Recall: Domain Shifts

- Source dataset: Cityscapes
- Target dataset: Foggy Cityscapes
- Network: DeepLabv2

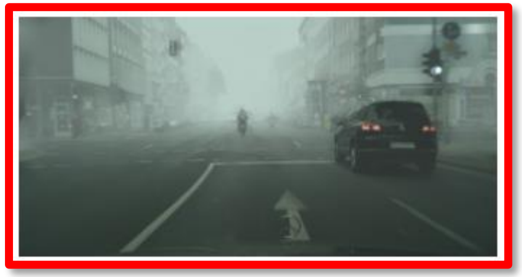
Cityscapes



Source domain



Foggy Cityscapes



Target domain

Robust Computer Vision

- Others

Adversarial Examples

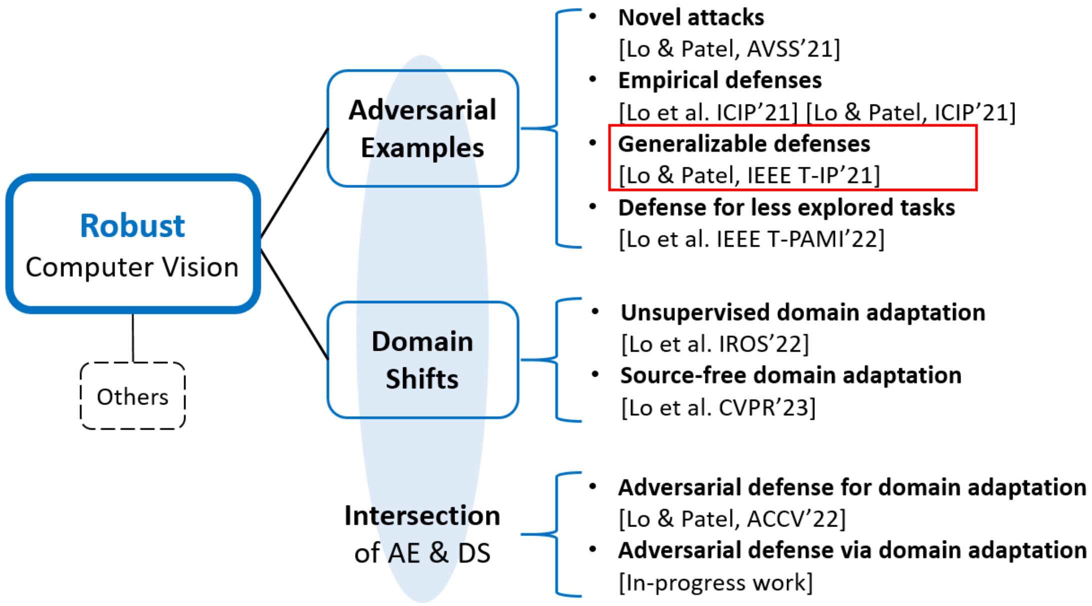
- **Novel attacks**
  [Lo & Patel, AVSS’21]
- **Empirical defenses**
  [Lo et al. ICIP’21] [Lo & Patel, ICIP’21]
- **Generalizable defenses**
  [Lo & Patel, IEEE T-IP’21]
- **Defense for less explored tasks**
  [Lo et al. IEEE T-PAMI’22]

Domain Shifts

- **Unsupervised domain adaptation**
  [Lo et al. IROS’22]
- **Source-free domain adaptation**
  [Lo et al. CVPR’23]

Intersection of AE & DS

- **Adversarial defense for domain adaptation**
  [Lo & Patel, ACCV’22]
- **Adversarial defense via domain adaptation**
  [In-progress work]



Defending Against Multiple and Unforeseen Adversarial Videos

Shao-Yuan Lo, *Student Member, IEEE* and Vishal M. Patel, *Senior Member, IEEE*

IEEE Transactions on Image Processing (T-IP), 2021

Adversarial Video Types

- PGD:
  Projective gradient descent
  [Madry et al. ICLR'18]

- ROA:
  Rectangular occlusion
  [Wu et al. ICLR'20]

- AF:
  Adversarial Framing
  [Zajac et al. AAAI'19]

- SPA:
  Salt-and-Pepper noise

How to simultaneously defend against multiple types of attacks?

Problem: Multi-perturbation Robustness

- Standard adversarial training has poor **multi-perturbation robustness**.
- Training: δ_{PGD}
- Test: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}

$$\underbrace{\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\underbrace{\max_{\delta\in\mathbb{S}} L(x+\delta, y; \theta)}_{\text{Generate one type of adversarial examples}} \right]}_{\text{Train model parameters}}$$

[Madry et al. ICLR'18]

Problem: Multi-perturbation Robustness

- **Average** adversarial training is better, but not enough.
- Training: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}
- Test: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}

$$\underbrace{\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\underbrace{\sum_{i=1}^{N} \max_{\delta_i \in \mathbb{S}_i} L(x+\delta_i, y; \theta)}_{\text{Generate multiple types of adversarial examples}} \right]}_{\text{Train model parameters}}$$

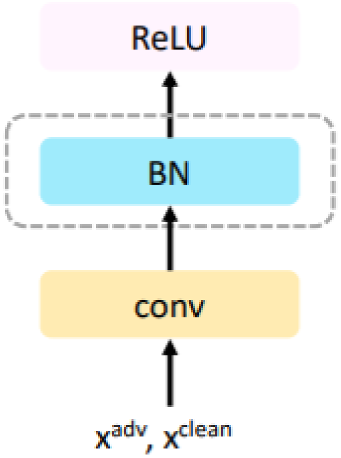
Generate **multiple types** of adversarial examples

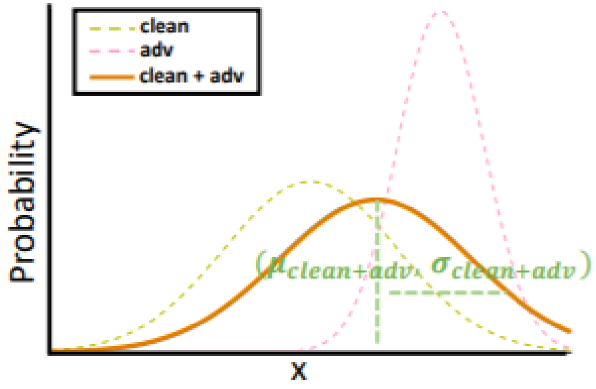
Train model parameters

[Tramèr & Boneh NeurIPS'19]

Observation: Distinct Data Distributions

- Why average adversarial training is **not** an ideal strategy?
- Example: **Clean** vs. **PGD**.
- Clean and PGD have **distinct** data distributions.
- The statistics estimation at **BN** may be confused when facing a mixture distribution.

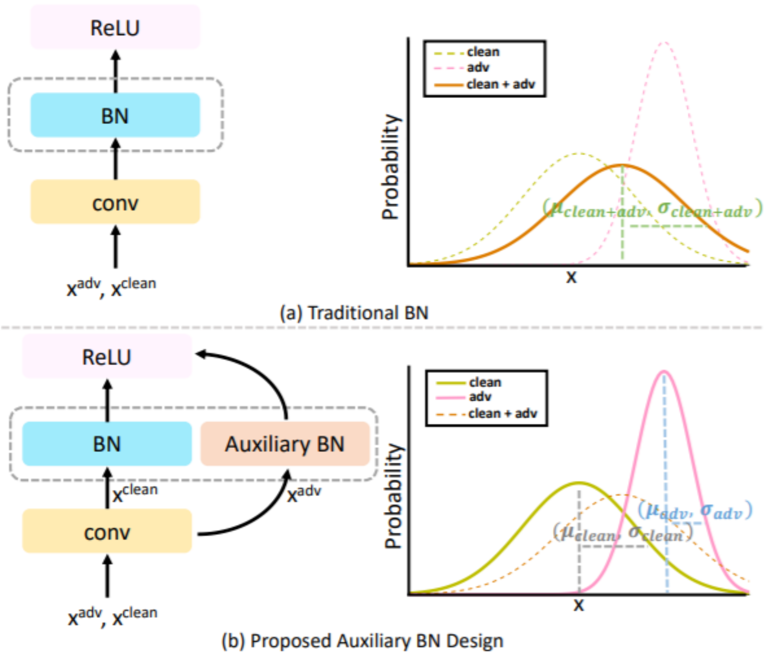




[Xie et al. CVPR'20]

Observation: Distinct Data Distributions

- Example: **Clean** vs. **PGD**.
- An **auxiliary BN** guarantees that data from different distributions are normalized separately.



(a) Traditional BN

(b) Proposed Auxiliary BN Design

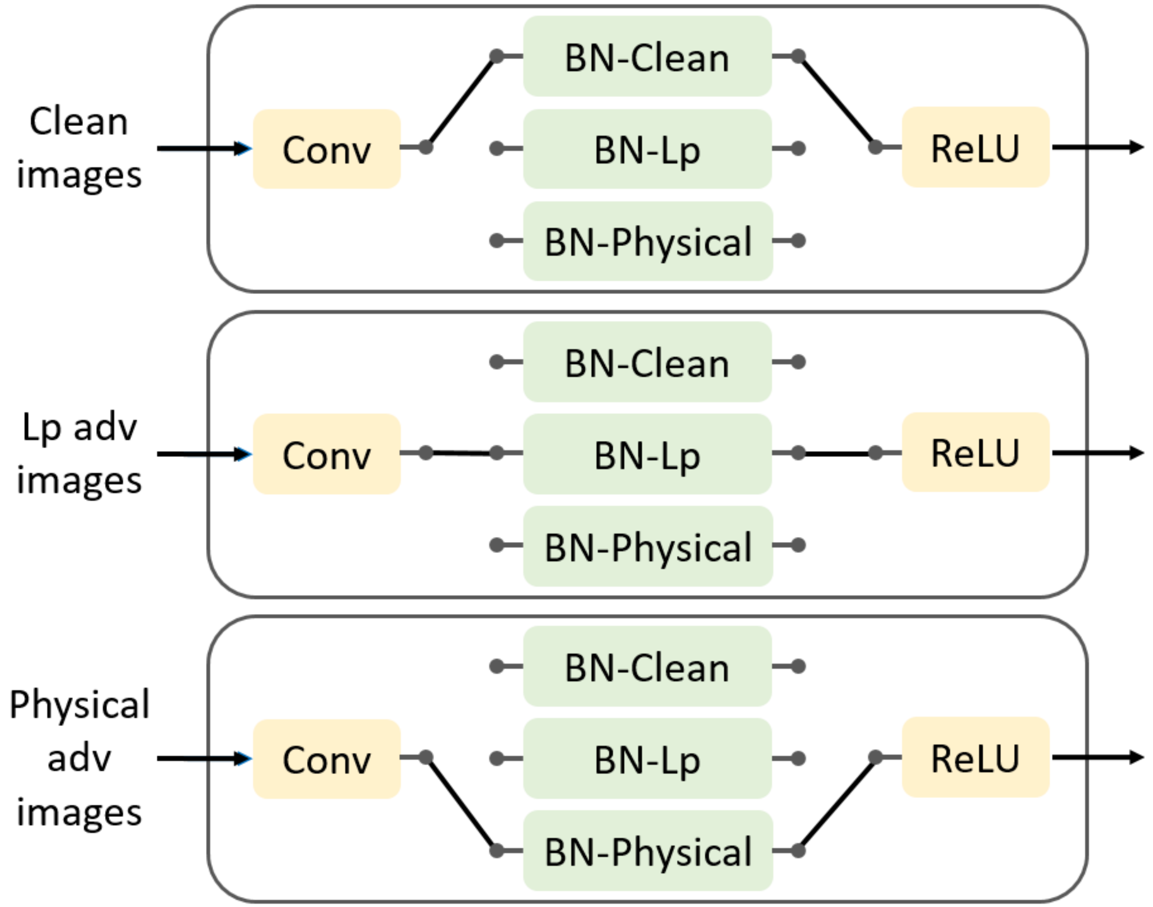
[Xie et al. CVPR'20]

Extension for Multi-perturbation Robustness

- What about **multiple** attack types (e.g., Clean, PGD, ROA, AF, SPA)?
- Our assumption: Different attack types have **distinct** data distributions.

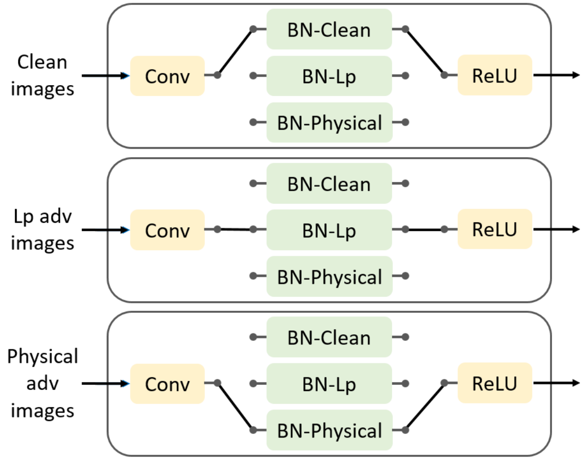
Our Solution: Multi-BN Structure

- Example:
  - **Known**: Clean, PGD, ROA
  - **Unforeseen**: AF, SPA
- **Lp-norm attacks**: PGD, SPA
- **Physically realizable attacks**: ROA, AF



Our Solution: Multi-BN Structure

- Training: Clean, δ_{PGD}, δ_{ROA}
- Test: Clean, δ_{PGD}, δ_{ROA}, δ_{AF}, δ_{SPA}

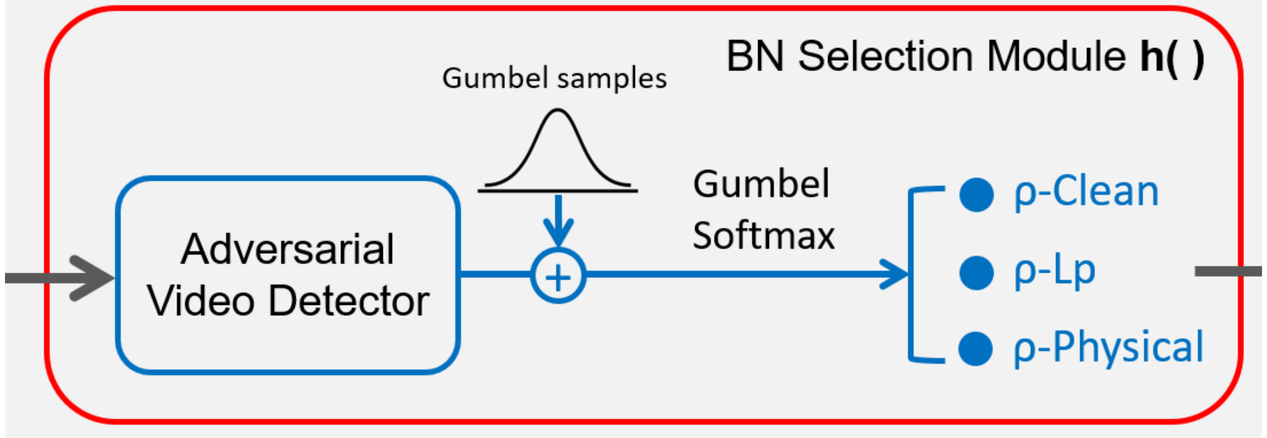


$$\theta = \theta^c + \sum_{i=0}^{N} \theta_i^b$$

$$\underbrace{\theta^* = \arg\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathbb{D}} \left[\underbrace{L(x, y; \theta^c, \theta_0^b)}_{\text{Clean data}} + \underbrace{\sum_{i=1}^{N} \max_{\delta_i \in \mathbb{S}_i} L(x + \delta_i, y; \theta^c, \theta_i^b)}_{\text{Generate multiple types of adversarial examples}} \right]}_{\text{Train model parameters}}$$

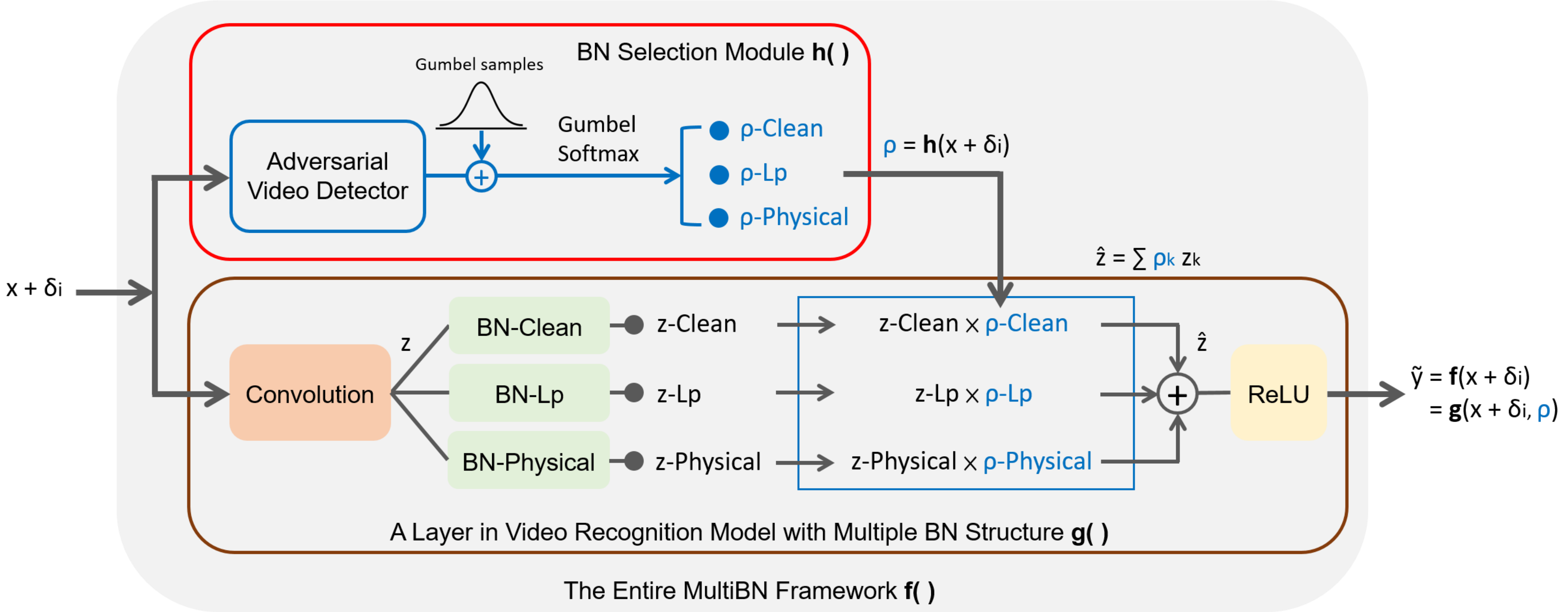
BN Selection Module

- At inference time, the input data have to pass through the corresponding BN branch **automatically**.
- The **adversarial video detector** is achieved by a video classifier.
- **Gumbel-Softmax** function [Jang et al. ICLR'17] is a **differentiable** approximation of the ***argmax*** operation.
- Use Gumbel-Softmax scores as ratio factors to weight each BN branch's output features.



Entire Framework

- End-to-end pipeline: $\tilde{y} = f(x + \delta_i; \theta^c, \theta^b, \theta^{det})$
$$= g(x + \delta_i, h(x + \delta_i; \theta^{det}); \theta^c, \theta^b)$$



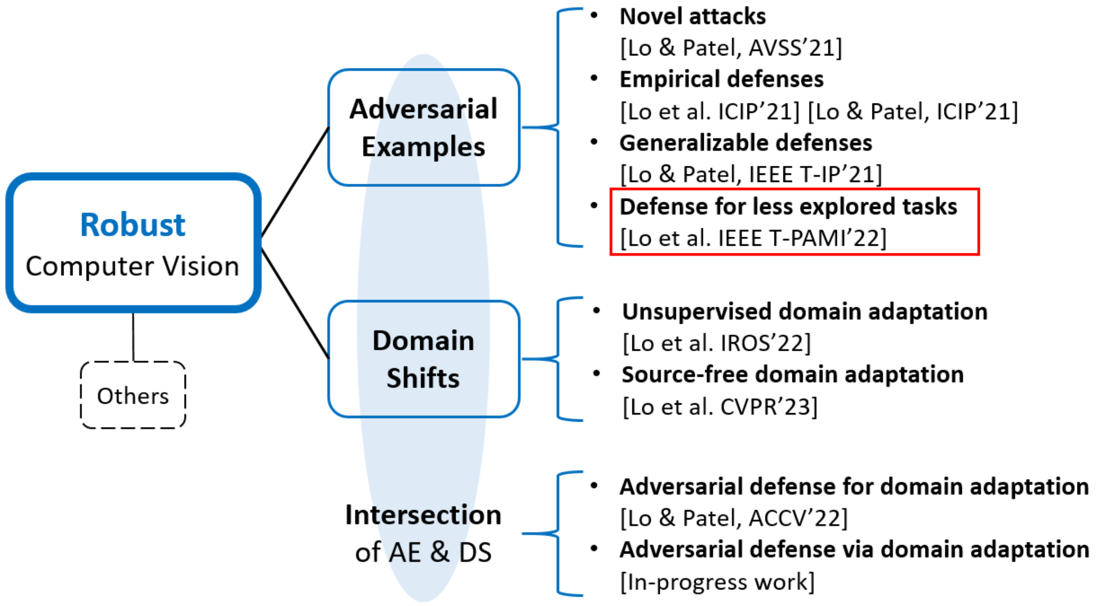
Results

Dataset: UCF-101

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**89.0**	3.3	0.5	1.6	8.4	20.6	0.0
TRADE [19] (ICML'19)	82.3	29.0	5.7	3.3	42.2	32.5	1.9
AVG [26] (NeurIPS'19)	68.9	38.1	51.4	18.5	49.6	45.3	17.3
MAX [26] (NeurIPS'19)	72.8	32.5	31.0	5.8	49.4	38.3	5.5
MSD [27] (ICML'20)	70.2	43.2	1.7	1.6	**56.0**	34.6	0.7
MultiBN (ours)	74.2	**44.6**	**58.6**	**44.3**	53.7	**55.1**	**34.8**

Dataset: HMDB-51

Model	Clean	PGD	ROA	AF	SPA	Mean	Union
No Defense	**65.1**	0.0	0.0	0.0	0.3	13.1	0.0
TRADE [19] (ICML'19)	54.8	6.8	0.3	0.0	20.5	16.5	0.0
AVG [26] (NeurIPS'19)	39.0	14.3	17.1	2.8	26.2	19.9	1.4
MAX [26] (NeurIPS'19)	48.6	13.9	16.0	0.1	30.3	21.8	0.0
MSD [27] (ICML'20)	41.4	18.2	0.1	0.0	**31.2**	18.2	0.0
MultiBN (ours)	51.1	**22.0**	**23.7**	**7.8**	29.9	**26.9**	**5.0**



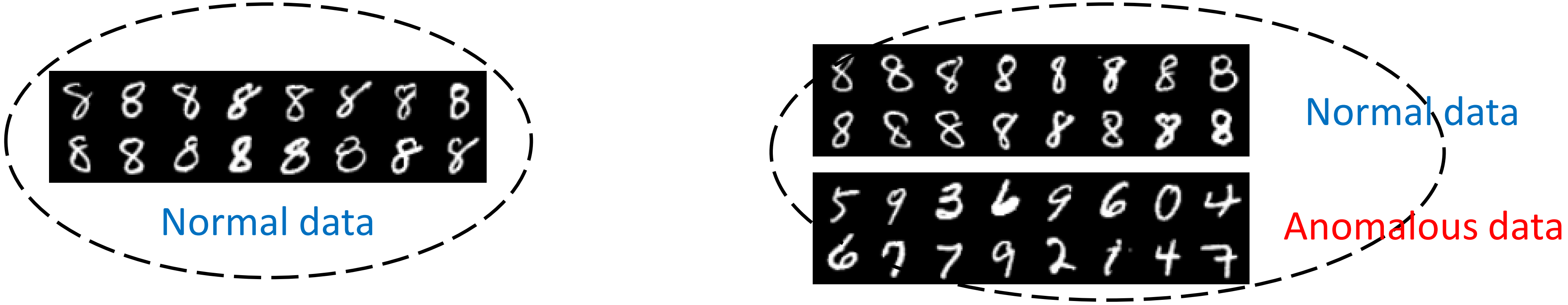
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2022

Adversarially Robust One-class Novelty Detection

Shao-Yuan Lo, *Student Member, IEEE,* Poojan Oza, *Student Member, IEEE,*
and Vishal M. Patel, *Senior Member, IEEE*

Recall: One-class Novelty Detection

- One-class novelty detection model is trained with examples of **a particular class** and is asked to identify whether a query example belongs to the same known class.
- Example:
  - **Known class** (normal data): 8
  - **Novel classes** (anomalous data): 0-7 & 9 (the rest of classes)

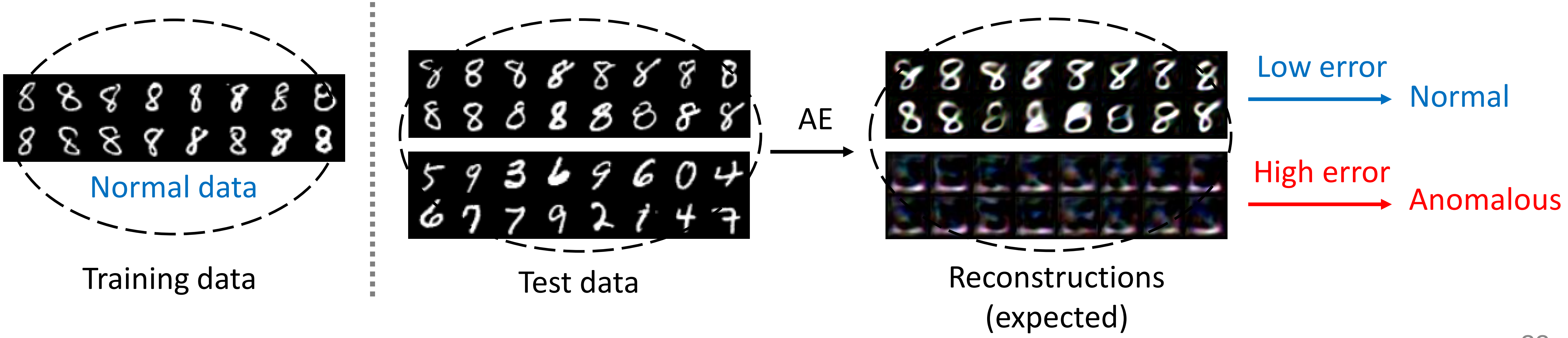


Training data

Test data

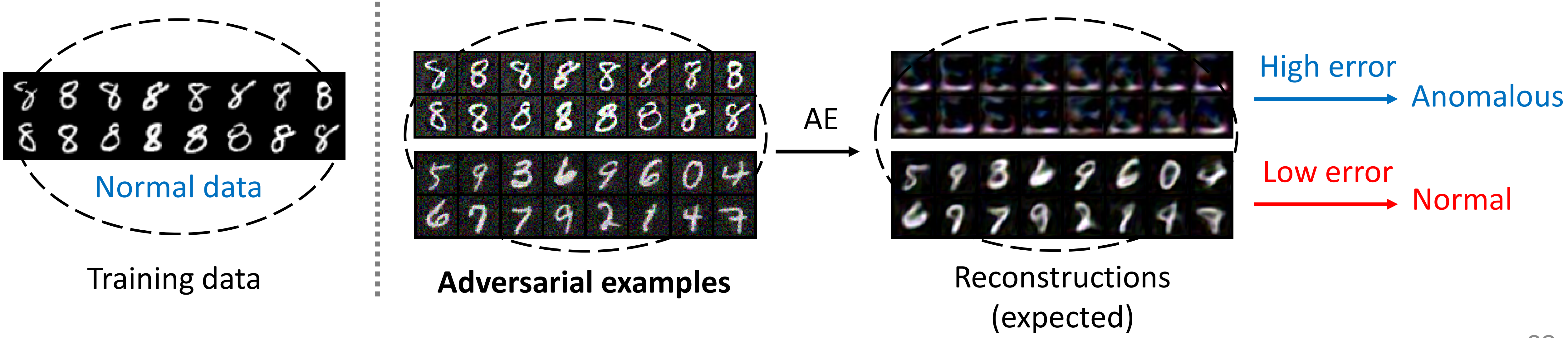
Recall: One-class Novelty Detection

- Most recent advances are based on the **autoencoder** architecture.
- Given an autoencoder that learns the distribution of the known class, we expect that the **normal data** are reconstructed accurately while the **anomalous data** are not.



Attacking One-class Novelty Detection

- How to generate adversarial examples against a novelty detector?
- If a test example is **normal**, **maximize** the reconstruction error.
- If a test example is **anomalous**, **minimize** the reconstruction error.

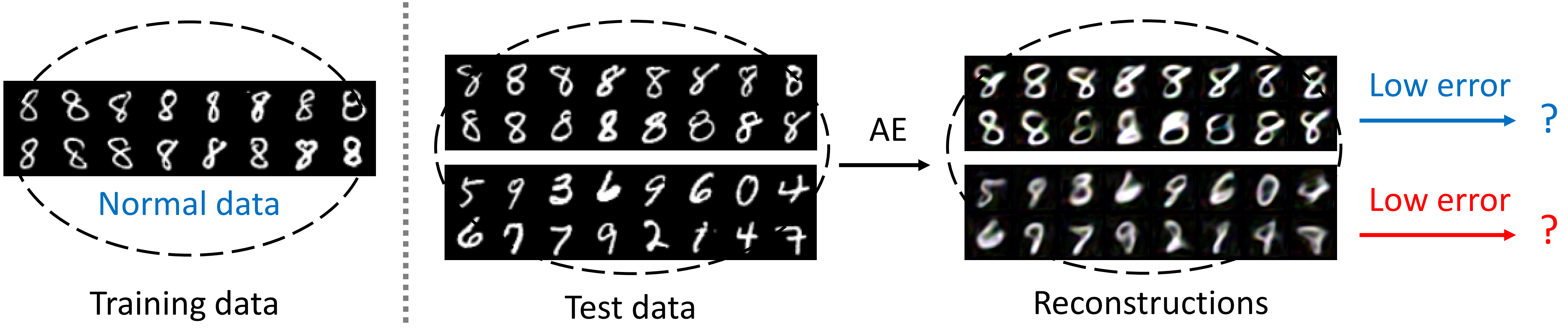


Goal: Adversarially Robust Novelty Detection

- Novelty detectors are **vulnerable** to adversarial attacks.
- Adversarially robust method specifically designed for novelty detectors is needed.
- A **new** research problem.

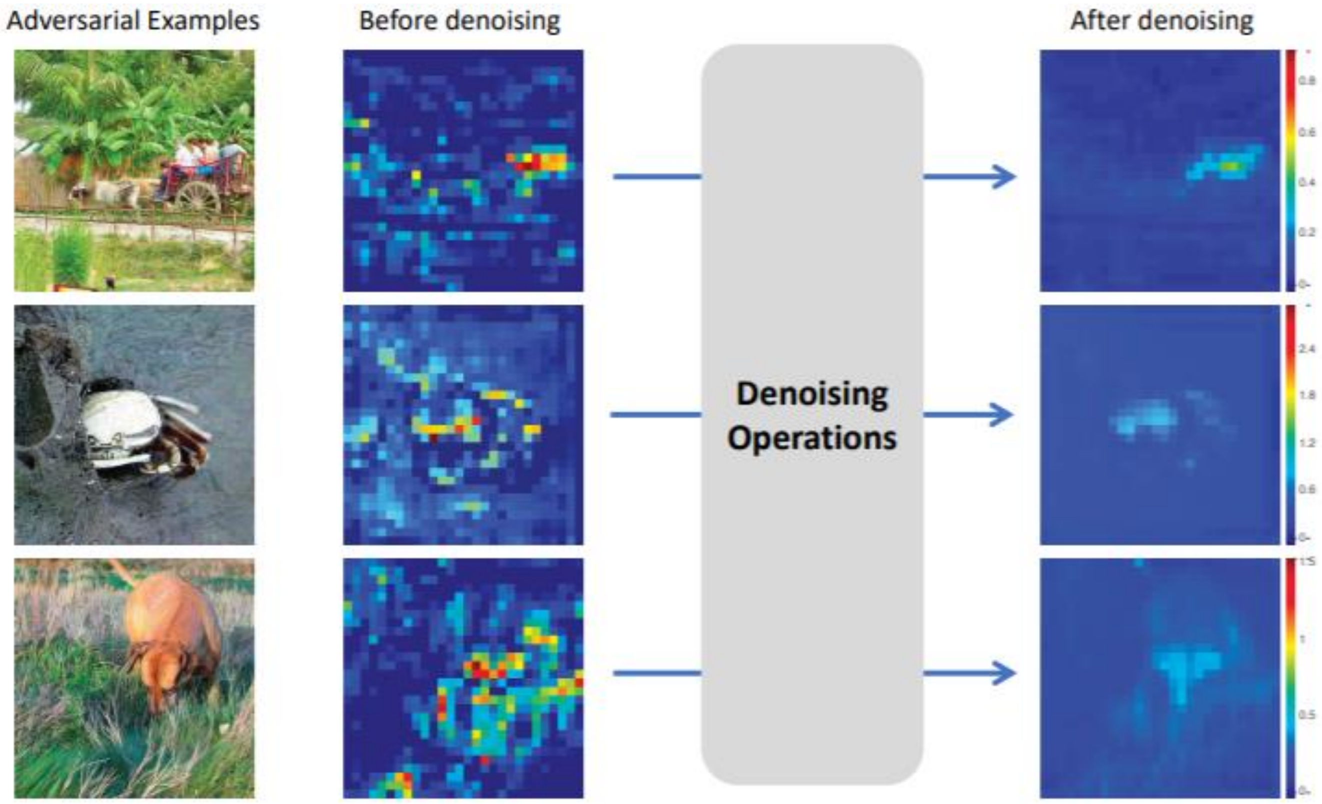
Observation: Generalizability

- Unique property: Preference for **poor** generalization of reconstruction ability.
- However, autoencoders have **good** generalizability.



Observation: Feature Denoising

- Adversarial perturbations can be removed in the **feature** domain.



[Xie et al. CVPR'19]

Our Solution

- **Observations**: Generalizability and Feature Denoising.

- **Assumption**: One can **largely** manipulate the latent space of a novelty detector to remove adversaries to a great extent, and this would not hurt the model capacity but **helps** if in a proper way.

- **Solution**: Learning **principal latent space**.

PCA Rephrased

- *h()* computes the **mean vector** and the first *k* **principal components** of the given data collection *X*:

$$h(\mathbf{X}, k) : \mathbf{X} \rightarrow \{\boldsymbol{\mu}, \tilde{\mathbf{U}}\}$$

- *f()* performs the forward PCA:

$$f(\mathbf{X}; \boldsymbol{\mu}, \tilde{\mathbf{U}}) = (\mathbf{X} - \boldsymbol{\mu}\mathbf{1}^\top)\tilde{\mathbf{U}}$$

$$\mathbf{X}_{pca} = f(\mathbf{X}; \boldsymbol{\mu}, \tilde{\mathbf{U}})$$

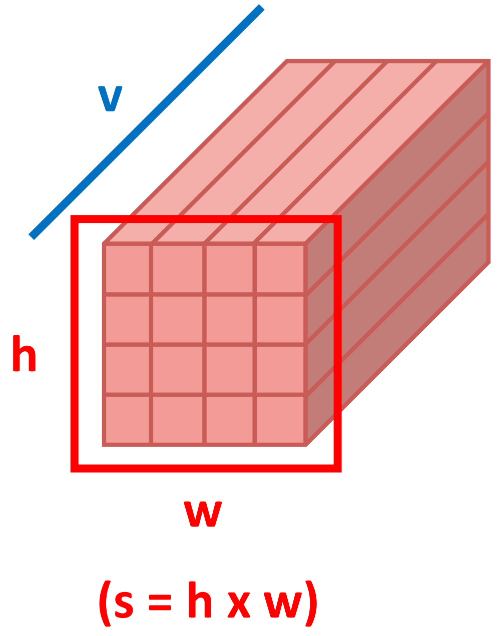
- *g()* performs the inverse PCA:

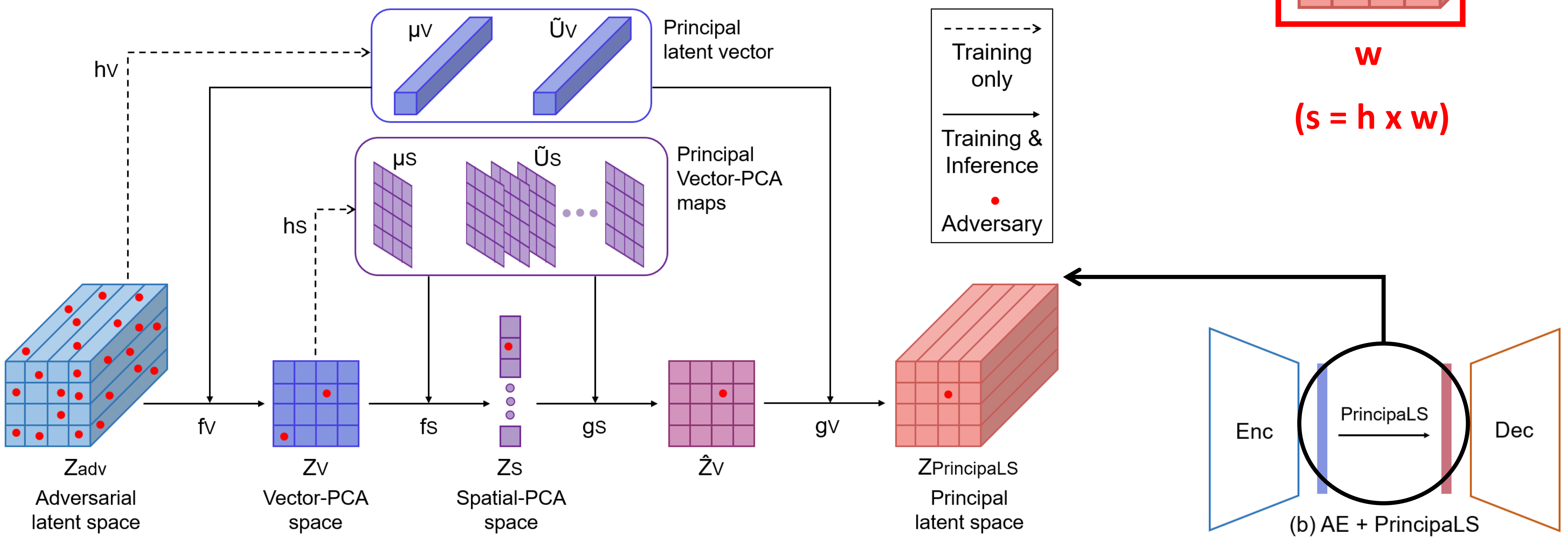
$$g(\mathbf{X}_{pca}; \boldsymbol{\mu}, \tilde{\mathbf{U}}) = \mathbf{X}_{pca}\tilde{\mathbf{U}}^\top + \boldsymbol{\mu}\mathbf{1}^\top$$

$$\hat{\mathbf{X}} = g(f(\mathbf{X}; \boldsymbol{\mu}, \tilde{\mathbf{U}}); \boldsymbol{\mu}, \tilde{\mathbf{U}})$$

Cascade PCA Process

- **Vector-PCA** performs PCA on the **vector** dimension.
- **Spatial-PCA** performs PCA on the **spatial** dimension.





Cascade PCA Process

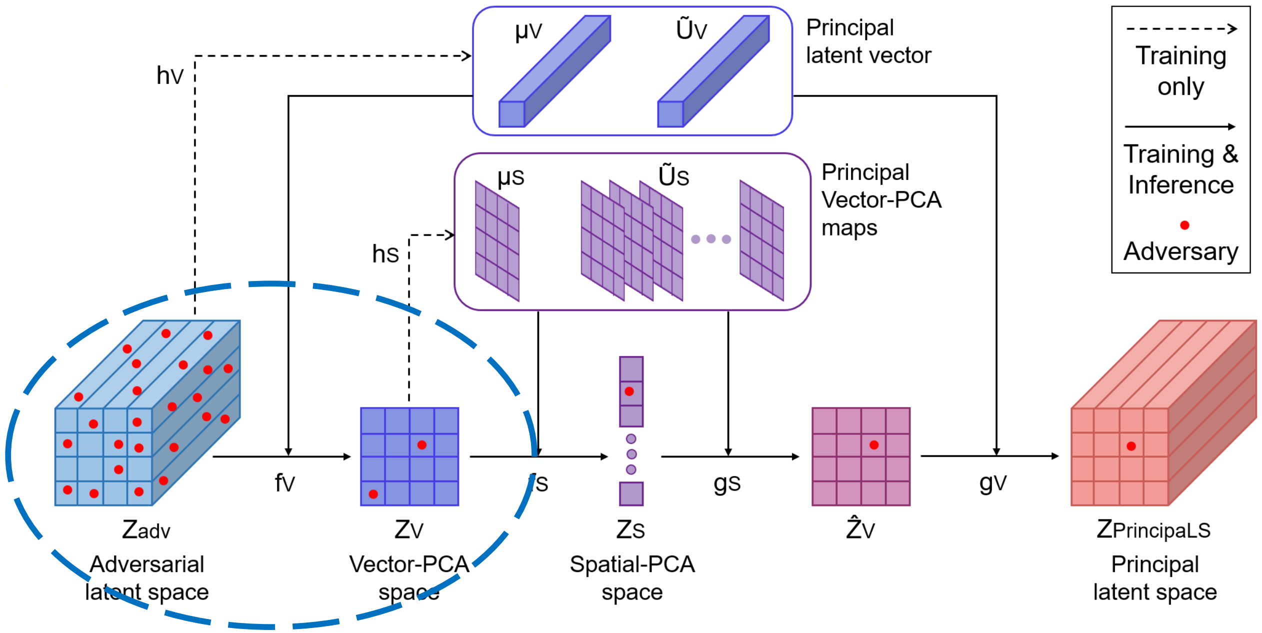
- Step 1: **Forward Vector-PCA**, i.e., *fv()*

$$\boldsymbol{Z}_{adv} \in \mathbb{R}^{s\times v} \longrightarrow \boldsymbol{Z}_{V} \in \mathbb{R}^{s\times 1}$$

Latent space — Vector-PCA space

$$\{\boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V\} = h_V(\mathbf{Z}, k_V = 1)$$

$$\mathbf{Z}_V = f_V(\mathbf{Z}; \boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V)$$



Cascade PCA Process

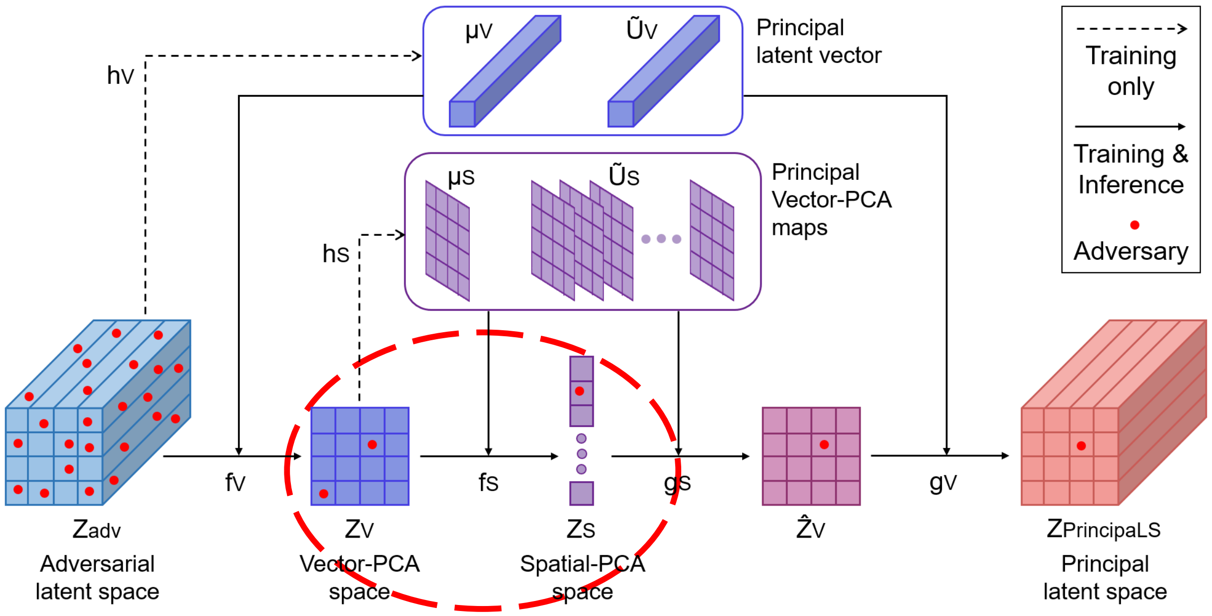
- Step 2: **Forward Spatial-PCA**, i.e., *fs()*

$$\boldsymbol{Z}_V \in \mathbb{R}^{s\times 1} \longrightarrow \boldsymbol{Z}_S \in \mathbb{R}^{k_S\times 1}$$

Vector-PCA space → Spatial-PCA space

$$\{\boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S\} = h_S(\mathbf{Z}_V^\top, k_S)$$

$$\mathbf{Z}_S^\top = f_S(\mathbf{Z}_V^\top; \boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S)$$



Cascade PCA Process

- Step 3: **Inverse Spatial-PCA**, i.e., *gs()*
- Step 4: **Inverse Vector-PCA**, i.e., *gv()*

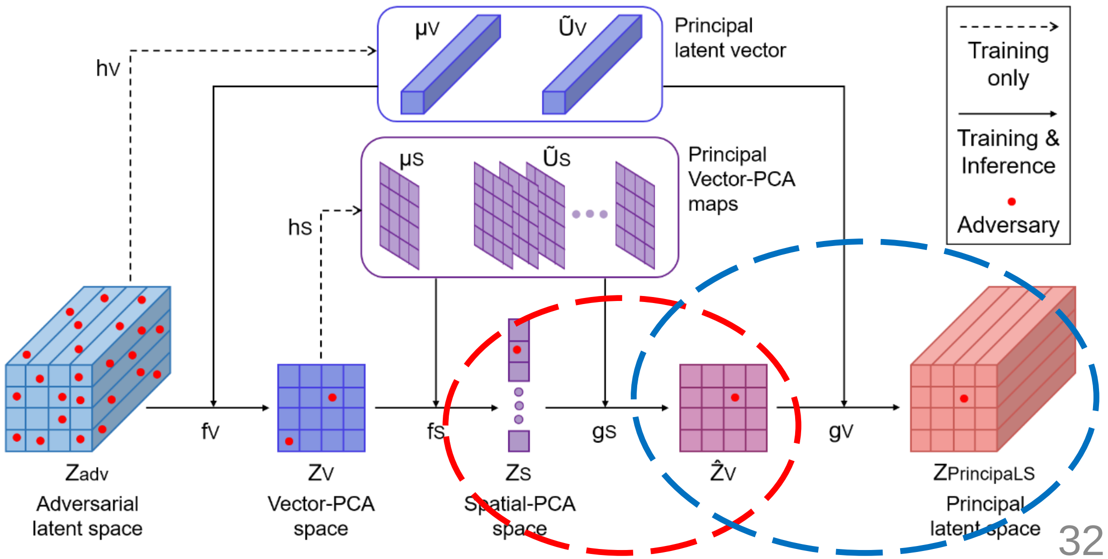
$$\boldsymbol{Z}_S \in \mathbb{R}^{k_S \times 1} \longrightarrow \boldsymbol{Z}_{pls} \in \mathbb{R}^{s \times v}$$

Spatial-PCA space

Principal latent space

$$\hat{\mathbf{Z}}_V^\top = g_S(\mathbf{Z}_S^\top; \boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S)$$

$$\mathbf{Z}_{plr} = g_V(\hat{\mathbf{Z}}_V; \boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V)$$



Learning Principal Latent Components

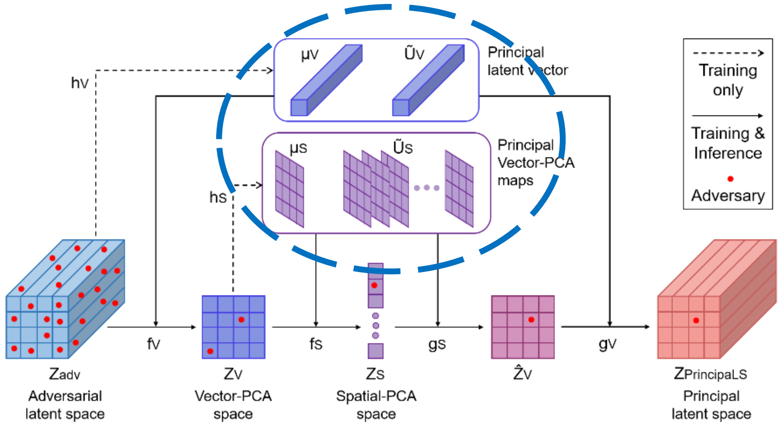
- **Principal latent components**:

$$\{\boldsymbol{\mu}_V, \tilde{\mathbf{U}}_V, \boldsymbol{\mu}_S, \tilde{\mathbf{U}}_S\}$$

- **Training time**: Train along with the network weights by exponential moving average (EMA).

$$\{\boldsymbol{\mu}_V^t, \tilde{\mathbf{U}}_V^t\} = \{\boldsymbol{\mu}_V^{t-1}, \tilde{\mathbf{U}}_V^{t-1}\} + \eta_V(h_V(\mathbf{Z}^t) - \{\boldsymbol{\mu}_V^{t-1}, \tilde{\mathbf{U}}_V^{t-1}\})$$

$$\{\boldsymbol{\mu}_S^t, \tilde{\mathbf{U}}_S^t\} = \{\boldsymbol{\mu}_S^{t-1}, \tilde{\mathbf{U}}_S^{t-1}\} + \eta_S(h_S(\mathbf{Z}^t) - \{\boldsymbol{\mu}_S^{t-1}, \tilde{\mathbf{U}}_S^{t-1}\})$$

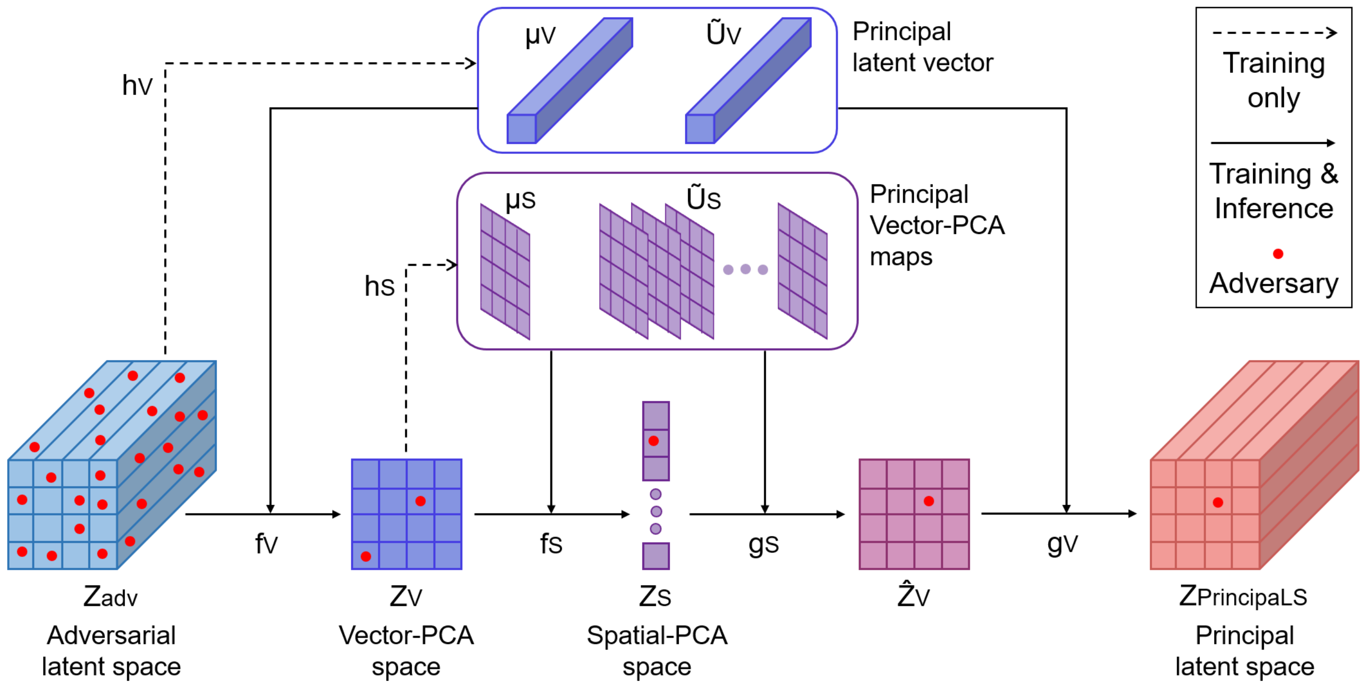


- **Inference time**: Perform the cascade PCA process with the fixed and well-trained parameters:

$$\{\boldsymbol{\mu}_V^*, \tilde{\mathbf{U}}_V^*, \boldsymbol{\mu}_S^*, \tilde{\mathbf{U}}_S^*\}$$

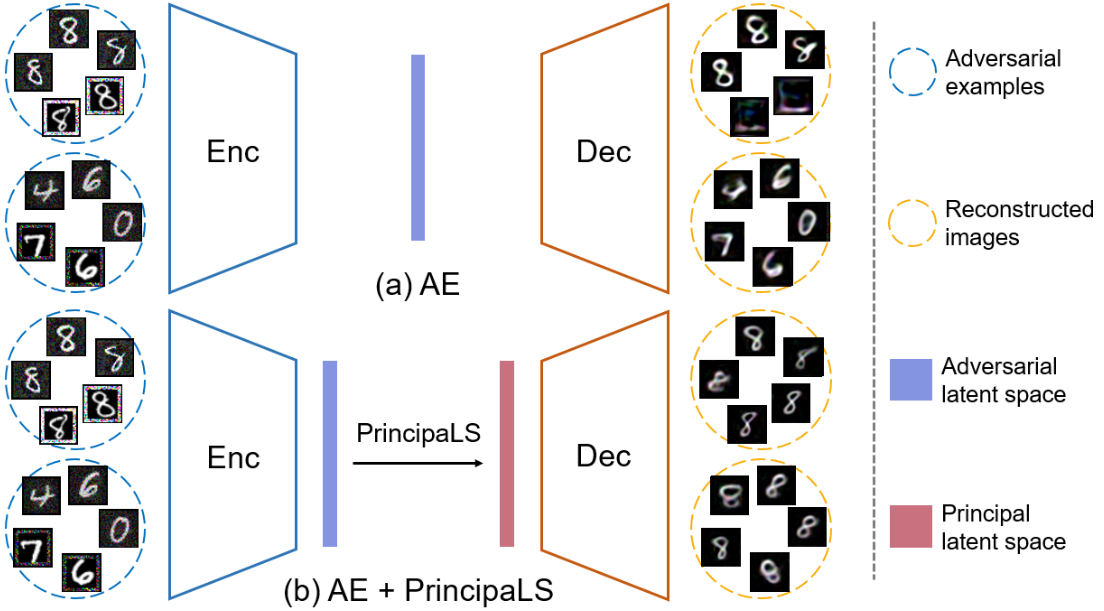
Defense Mechanism

- **Vector-PCA** replaces the perturbed latent vectors with the clean principal latent vector.
- **Spatial-PCA** removes the remaining perturbations on the Vector-PCA map.



Defense Mechanism

- Combine **adversarial training**.
- The proposed PrincipaLS process can robustify **any** AE-based novelty detectors.
  - AE, VAE, AAE, ALOCC (CVPR'18), GPND (NeurIPS'18), etc.



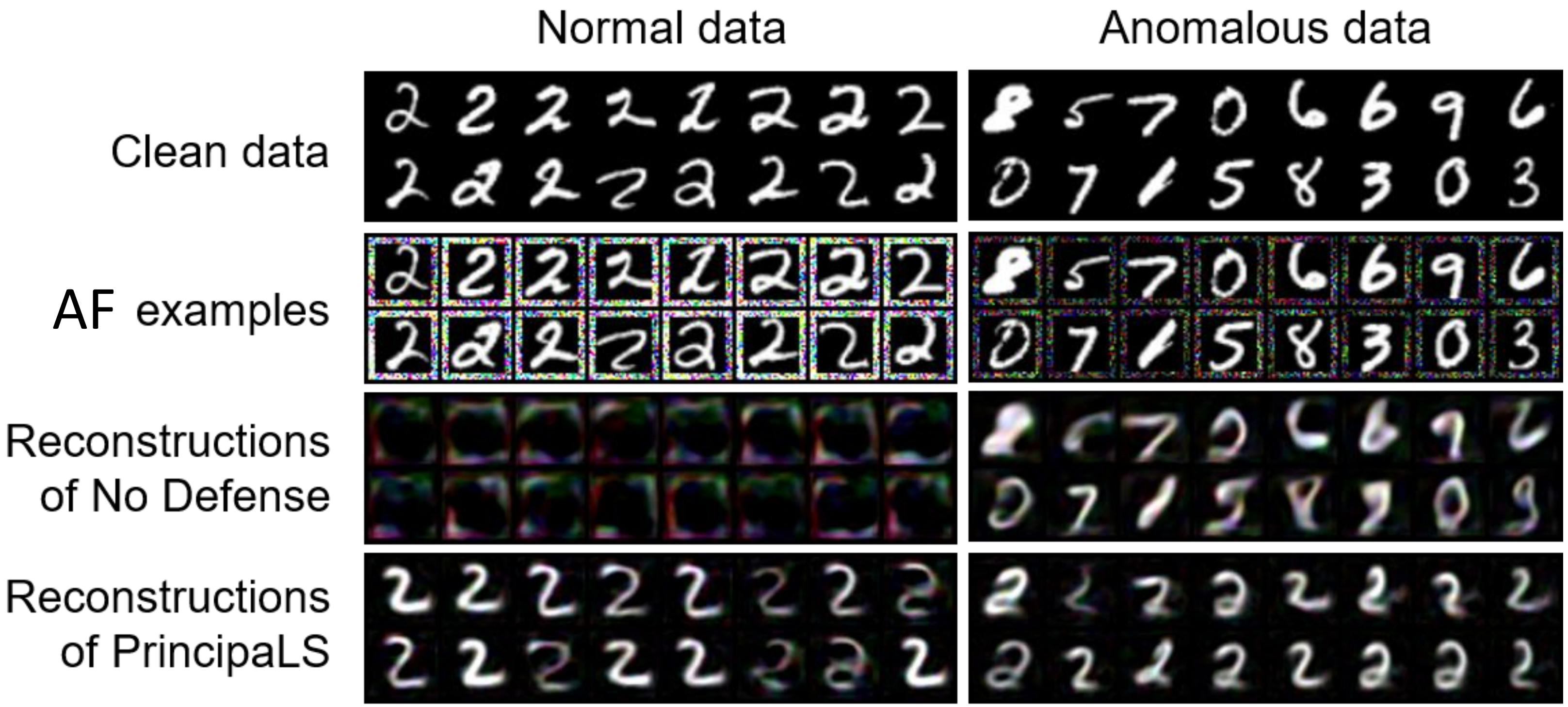
Results

- Evaluation metric: mean of AUROC
- PrincipaLS is effective on **5** datasets against **6** attacks for **7** novelty detection methods.

Dataset	Defense	Clean	FGSM [11]	PGD [27]	MI-FGSM [36]	MultAdv [37]	AF [38]	Black-box [47]	Average
MNIST [48]	No Defense	0.964	0.350	0.051	0.022	0.170	0.014	0.790	0.337
	PGD-AT [27]	0.961	0.604	0.357	0.369	0.444	0.155	0.691	0.512
	FD [15]	0.963	0.612	0.366	0.379	0.453	0.142	0.700	0.516
	SAT [23]	0.947	0.527	0.295	0.306	0.370	0.142	0.652	0.463
	RotNet-AT [21]	0.967	0.598	0.333	0.333	0.424	0.101	0.695	0.493
	SOAP [22]	0.940	0.686	0.504	0.506	0.433	0.088	0.863	0.574
	APAE [46]	0.925	0.428	0.104	0.105	0.251	0.022	0.730	0.366
	PrincipaLS (ours)	**0.973**	**0.812**	**0.706**	**0.707**	**0.725**	**0.636**	**0.866**	**0.775**
SHTech [52]	No Defense	0.523	0.204	0.034	0.038	0.006	0.000	0.220	0.146
	PGD-AT [27]	0.527	0.217	0.168	0.154	0.100	0.000	0.221	0.198
	FD [15]	0.528	0.226	0.189	0.181	0.132	0.002	0.229	0.212
	SAT [23]	**0.529**	0.184	0.110	0.092	0.040	0.000	0.199	0.165
	RotNet-AT [21]	0.516	0.220	0.163	0.158	0.113	0.000	0.229	0.200
	SOAP [22]	0.432	0.024	0.002	0.000	0.002	**0.181**	0.202	0.120
	APAE [46]	0.510	0.215	0.048	0.050	0.011	0.000	0.207	0.149
	PrincipaLS (ours)	0.498	**0.274**	**0.223**	**0.217**	**0.175**	0.051	**0.308**	**0.249**

Analysis

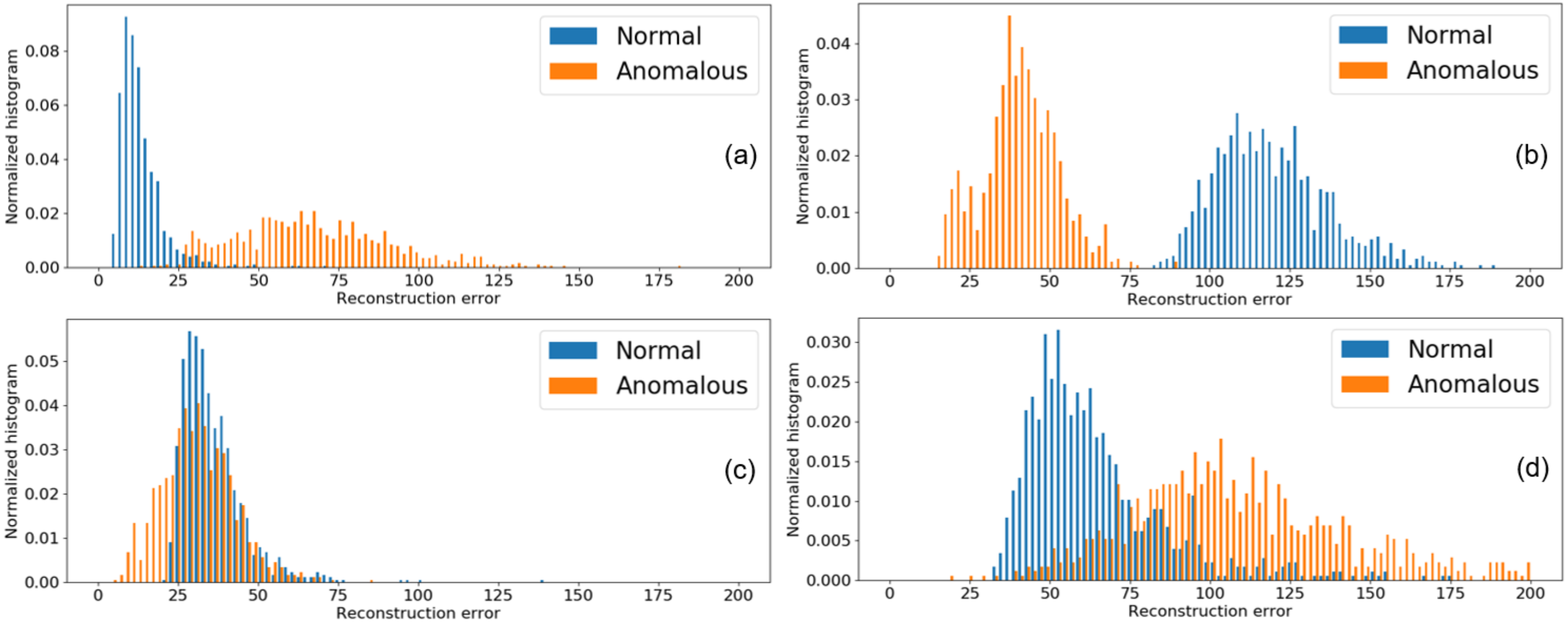
- PrincipaLS reconstructs **every** input example to the known class (digit 2).

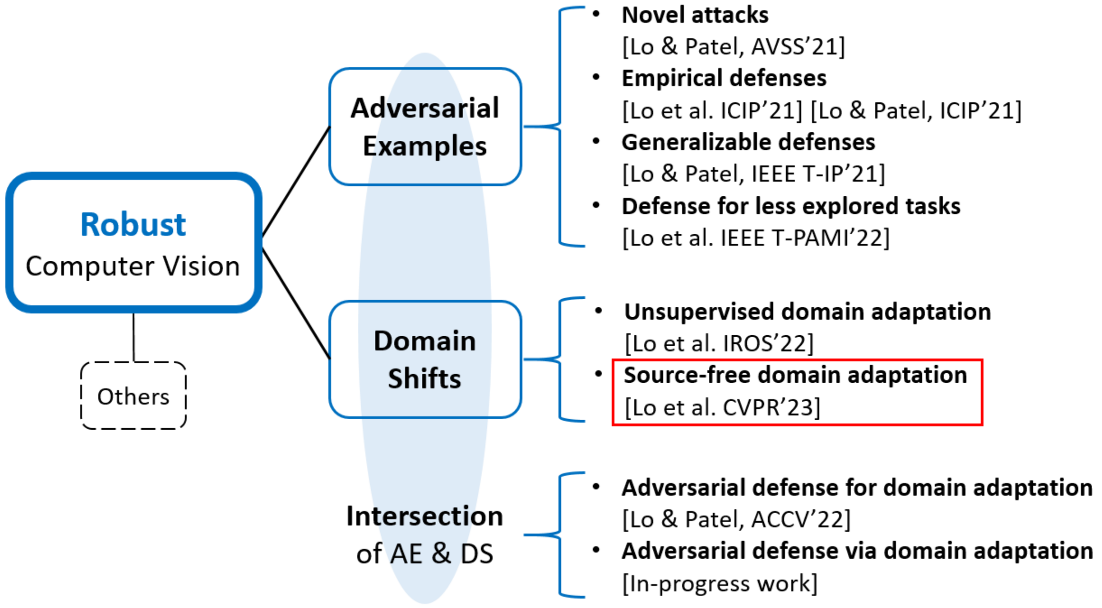


(b) AF attack

Analysis

- (a) No Defense under clean data (b) No Defense under PGD attack (c) PGD-AT under PGD attack (d) PrincipaLS under PGD attack
- PrincipaLS enlarges the reconstruction errors of anomalous data to a great extent.





Spatio-Temporal Pixel-Level Contrastive Learning-based Source-Free Domain Adaptation for Video Semantic Segmentation

Shao-Yuan Lo[1] Poojan Oza[2] Sumanth Chennupati[2] Alejandro Galindo[2] Vishal M. Patel[1]
[1]Johns Hopkins University [2]Amazon
{sylo, vpatel36}@jhu.edu {poojanku, sumchenn, pagh}@amazon.com

CVPR 2023

Recall: Video Semantic Segmentation

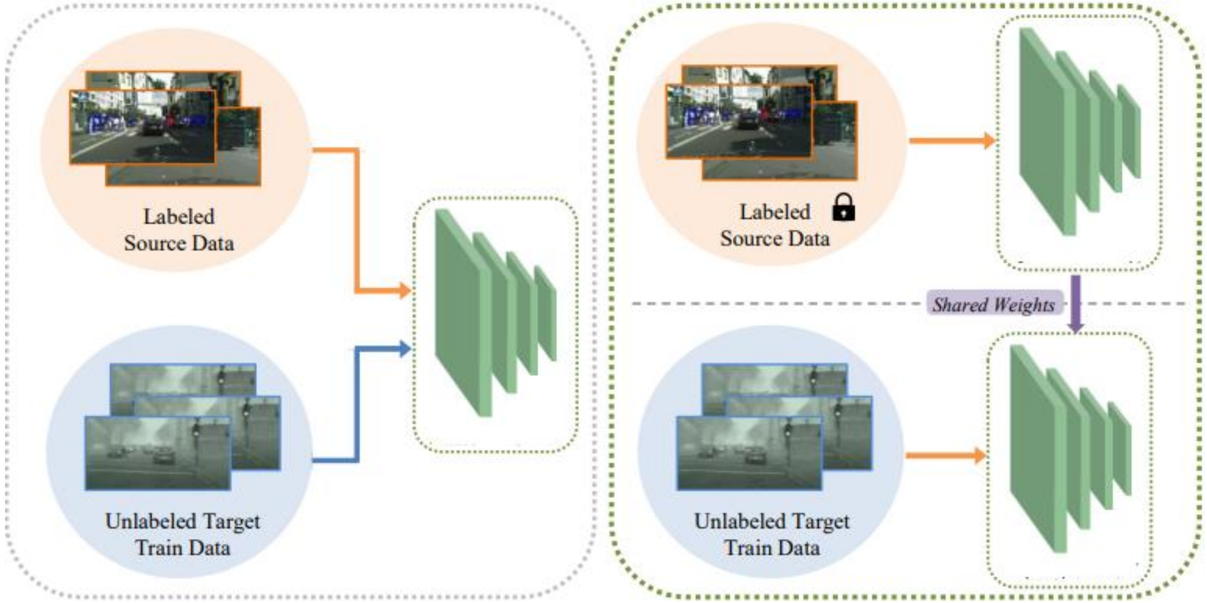
- Video semantic segmentation (VSS) aims to predict pixel-level semantics for each video frame.
- Compared to image semantic segmentation (ISS), temporal information can be exploited to improve either **accuracy** or **inference speed**.

[Jain et al. CVPR'19]

Source-Free Domain Adaptation

- **Scenario:** Training (source) and test (target) data are from different domains, and **we cannot access to the source data** (e.g. privacy).
- **Setting:** Given a **source-trained model** and an **unlabeled target dataset**, adapt the model to the **target** domain.

Classic domain adaptation



Source-free domain adaptation

[VS et al. WACV'23]

Challenges

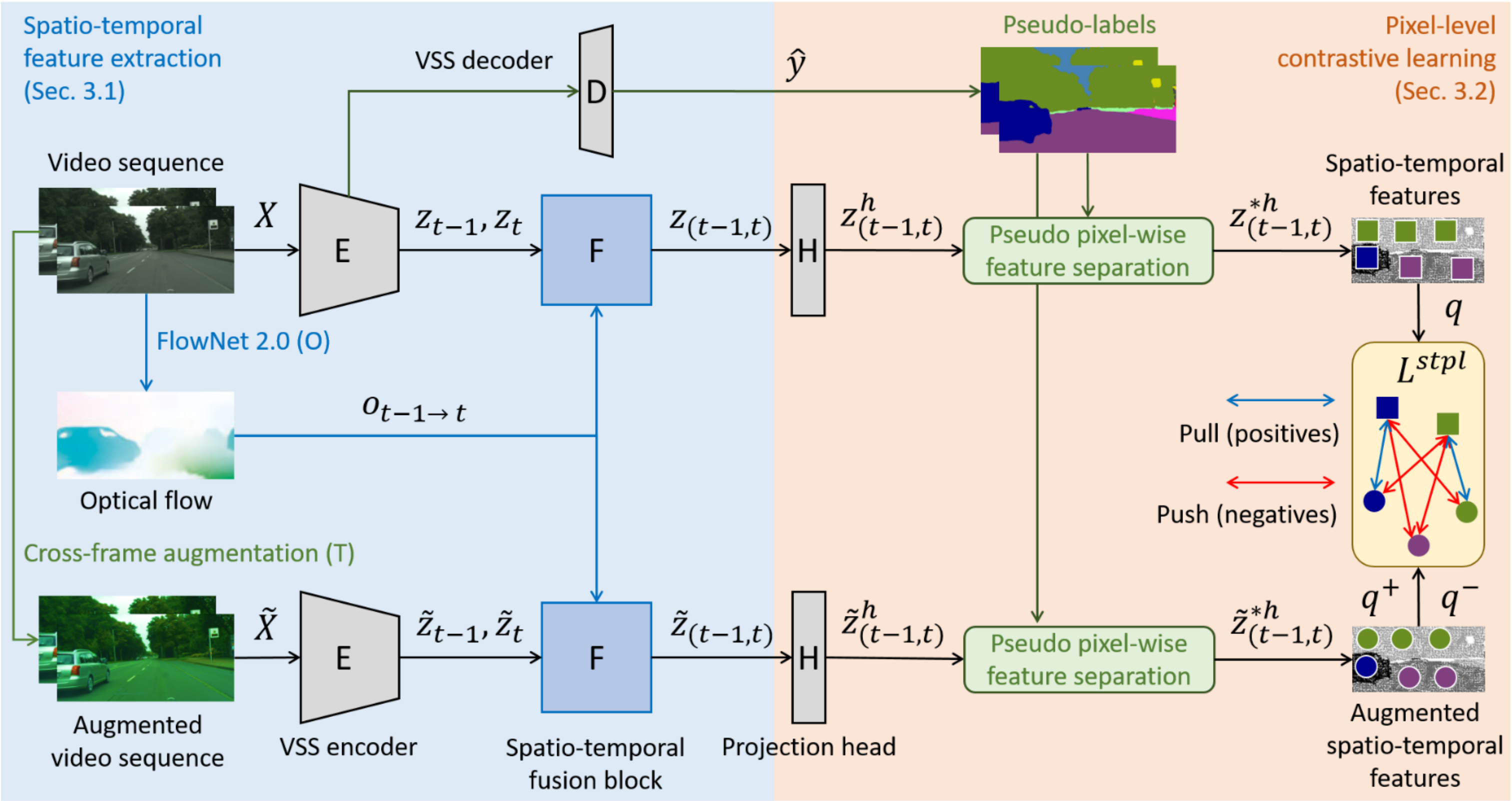
- Classic domain adaptation (UDA) for VSS methods are **not applicable** to the source-free domain adaptation (SFDA) setting.

- SFDA for ISS methods do not consider the **temporal information**.

- No access to any labeled training data.

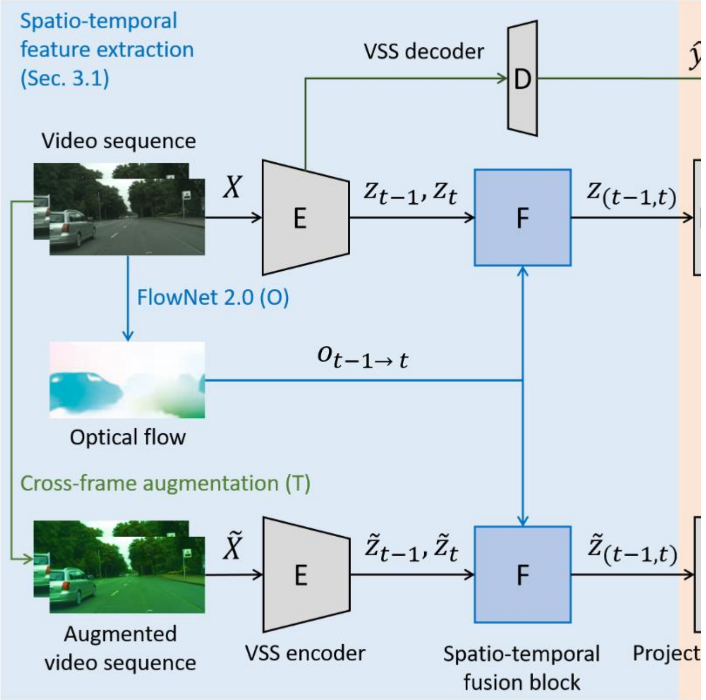
Spatio-Temporal Pixel-Level Contrastive Learning

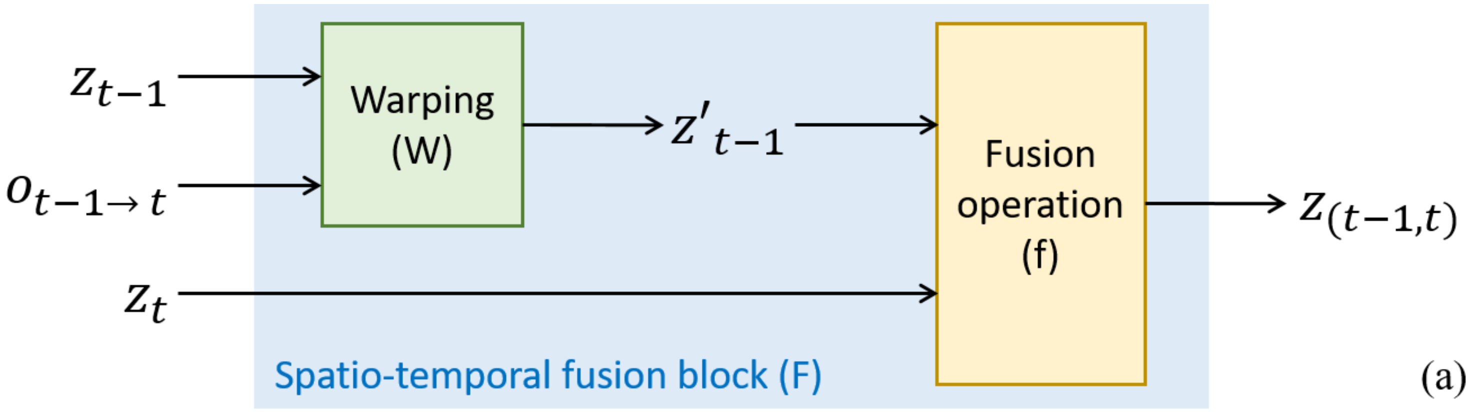
- Spatio-temporal feature extraction
- Pixel-level contrastive learning



Spatio-Temporal Feature Extraction

- Spatio-temporal fusion block
  - Feature warping by **optical flow** (temporal information)
  - Fusion operation: concatenation, element-wise addition, 1x1 convolution, attention module, etc.

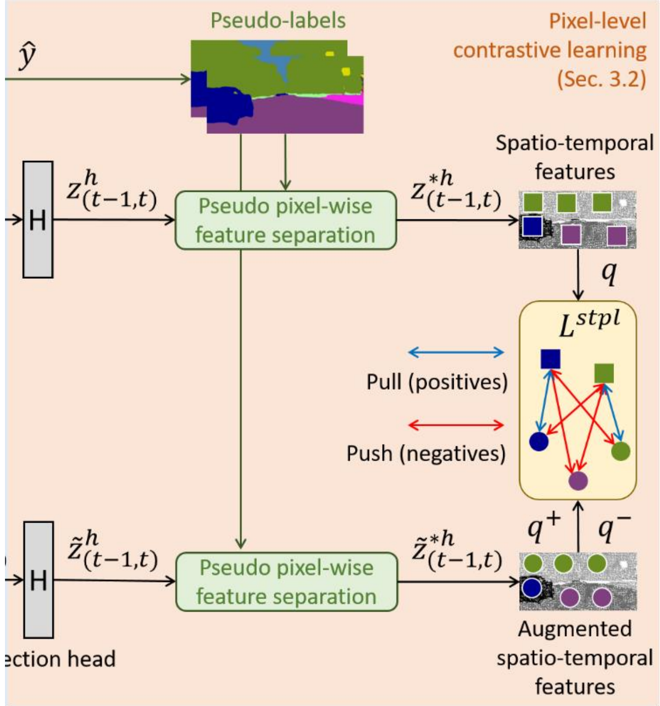




(a)

Pixel-Level Contrastive Learning

- Pseudo-labels are used for **pseudo pixel-wise feature separation**
- **Positive samples**: Pixels of the same semantic class
- **Negative samples**: Pixels of different semantic classes



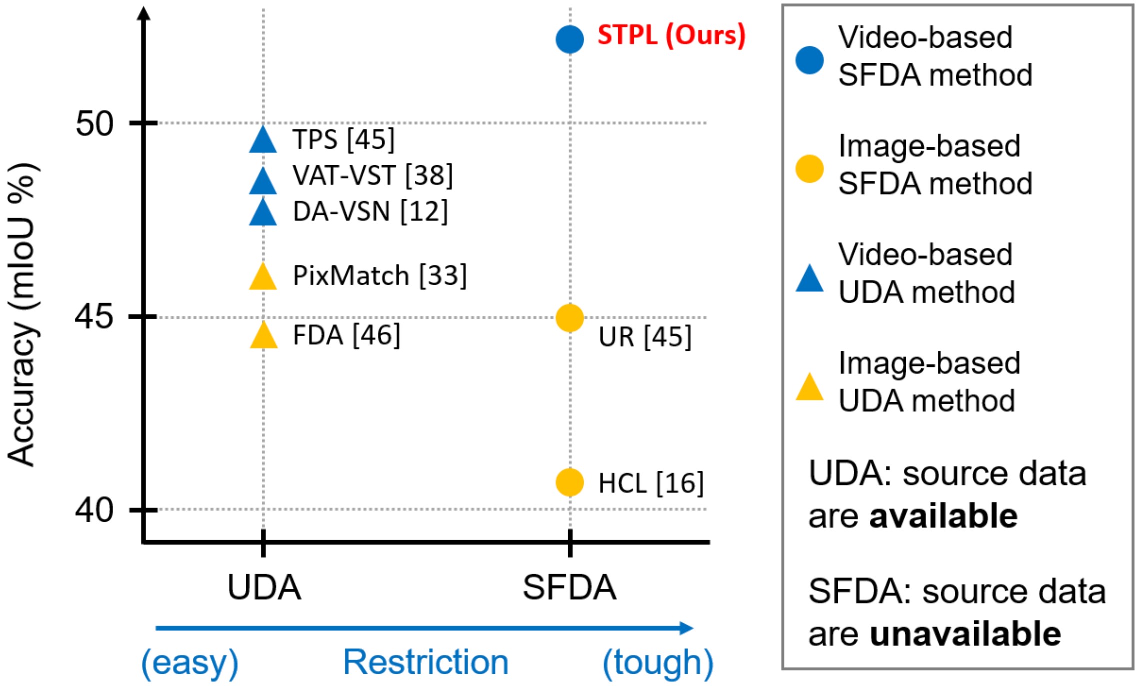
Pixel-wise SimCLR

$$\mathcal{L}_q^{stpl} = \frac{-1}{|P_q|} \sum_{q^+ \in P_q} \log \frac{\exp(q \cdot q^+ / \tau)}{\sum_{q^- \in N_q} \exp(q \cdot q^- / \tau)}$$

Results

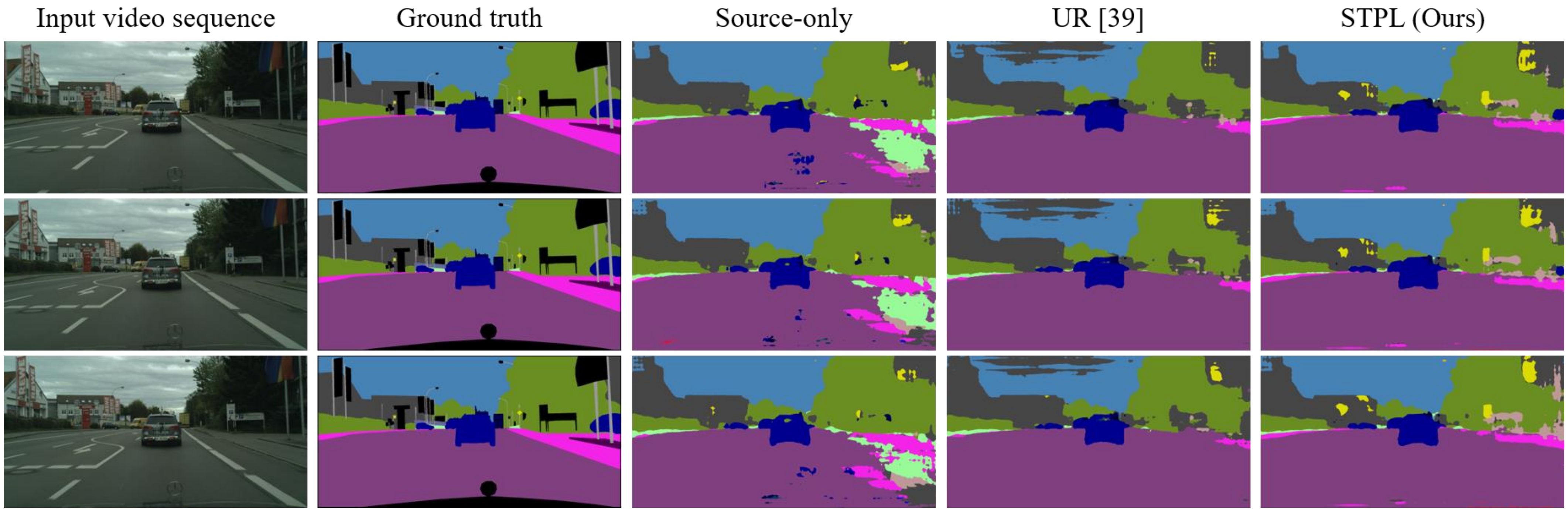
- Benchmark: VIPER → Cityscapes-Seq

Method	Design	DA	mIoU
Source-only	-	-	37.1
FDA [46] (CVPR'20)	Image	UDA	44.4
PixMatch [33] (CVPR'21)	Image	UDA	46.7
RDA [17] (ICCV'21)	Image	UDA	44.4
UR [39] (CVPR'21)	Image	SFDA	45.0
HCL [16] (NeurIPS'21)	Image	SFDA	41.5
DA-VSN [12] (ICCV'21)	Video	UDA	47.8
VAT-VST [38] (AAAI'22)	Video	UDA	48.7
TPS [45] (ECCV'22)	Video	UDA	48.9
DA-VSN* [12] (ICCV'21)	Video	SFDA	45.3
VAT-VST* [38] (AAAI'22)	Video	SFDA	43.6
TPS* [45] (ECCV'22)	Video	SFDA	27.8
STPL (Ours)	Video	SFDA	52.5
Oracle	-	-	69.9



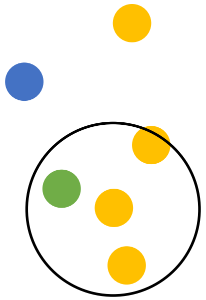
Results

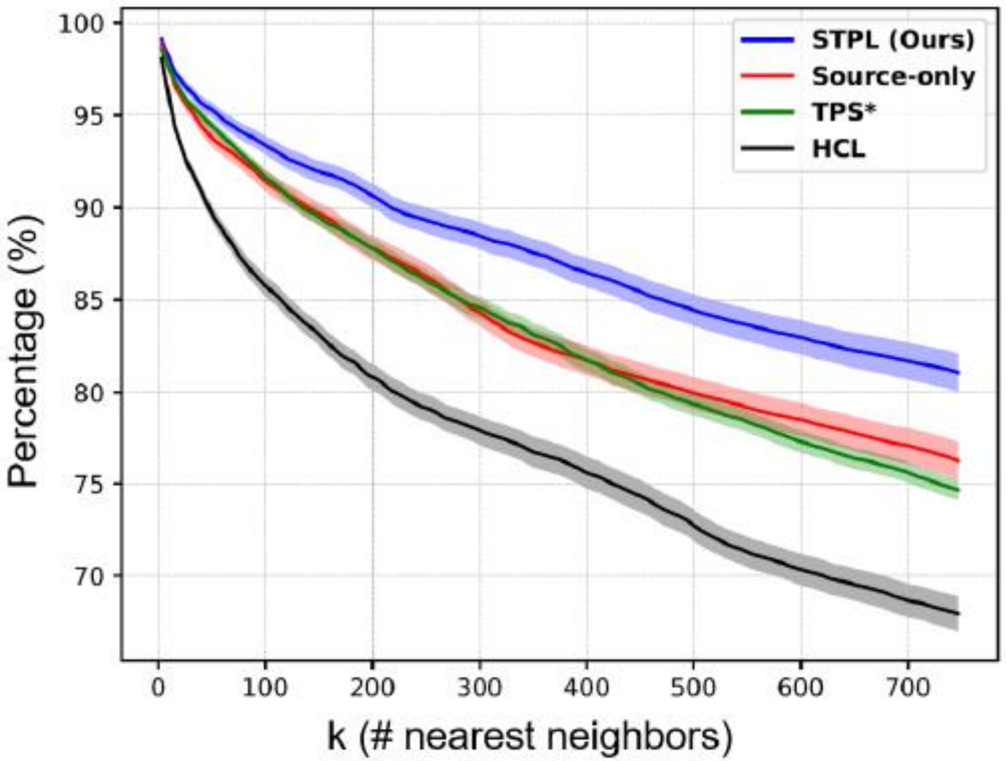
- Benchmark: VIPER → Cityscapes-Seq



Analysis

- The percentage of same-class pixel representations among the k-nearest neighbors in the feature space.





Robust
Computer Vision

- **Adversarial Examples**
  - **Novel attacks**
    [Lo & Patel, AVSS’21]
  - **Empirical defenses**
    [Lo et al. ICIP’21] [Lo & Patel, ICIP’21]
  - **Generalizable defenses**
    [Lo & Patel, IEEE T-IP’21]
  - **Defense for less explored tasks**
    [Lo et al. IEEE T-PAMI’22]
- **Domain Shifts**
  - **Unsupervised domain adaptation**
    [Lo et al. IROS’22]
  - **Source-free domain adaptation**
    [Lo et al. CVPR’23]
- Others

Intersection of AE & DS
- **Adversarial defense for domain adaptation**
  [Lo & Patel, ACCV’22]
- **Adversarial defense via domain adaptation**
  [In-progress work]


Prof. Vishal M. Patel


Poojan Oza


Jeya Maria Jose


Shaoyan Pan


Wei Wang


Jim Thomas


Cheng-Hao Kuo


Jingjing Zheng


Sumanth Chennupati


Alejandro Galindo

DARPA

amazon