# Memorizing Normality to Detect Anomaly: MemAE



## Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection

Dong Gong[1], Lingqiao Liu[1], Vuong Le[2], Budhaditya Saha[2],
Moussa Reda Mansour[3], Svetha Venkatesh[2], Anton van den Hengel[1]
[1]The University of Adelaide, Australia [2]A2I2, Deakin University [3]University of Western Australia
https://donggong1.github.io/anomdec-memae

# Memorizing Normality to Detect Anomaly: MemAE

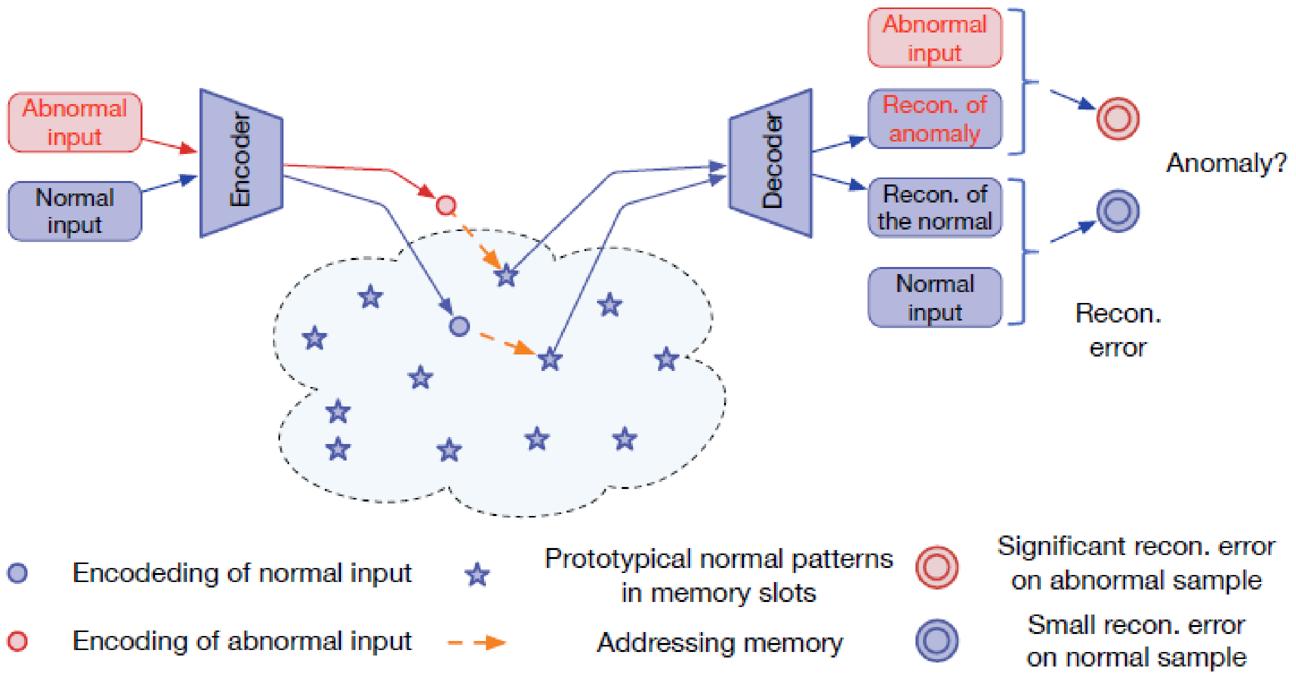
Anomaly Detection

1. Reconstruction-based
   - AE, VAE, GAN

2. Feature-based
   pre-trained CNN

# Memorizing Normality to Detect Anomaly: MemAE

## Concept




MemAE firstly obtains the encoding from the encoder and then uses it as a query to retrieve the most relevant memory items for reconstruction

# Memorizing Normality to Detect Anomaly: MemAE

1. Given an input, the encoder first obtains the encoding of the input.

2. By using the encoded representation as a query, the memory module retrieves the most relevant items in the memory via the **attention-based addressing operator**,

3. which are then delivered to the decoder for reconstruction.

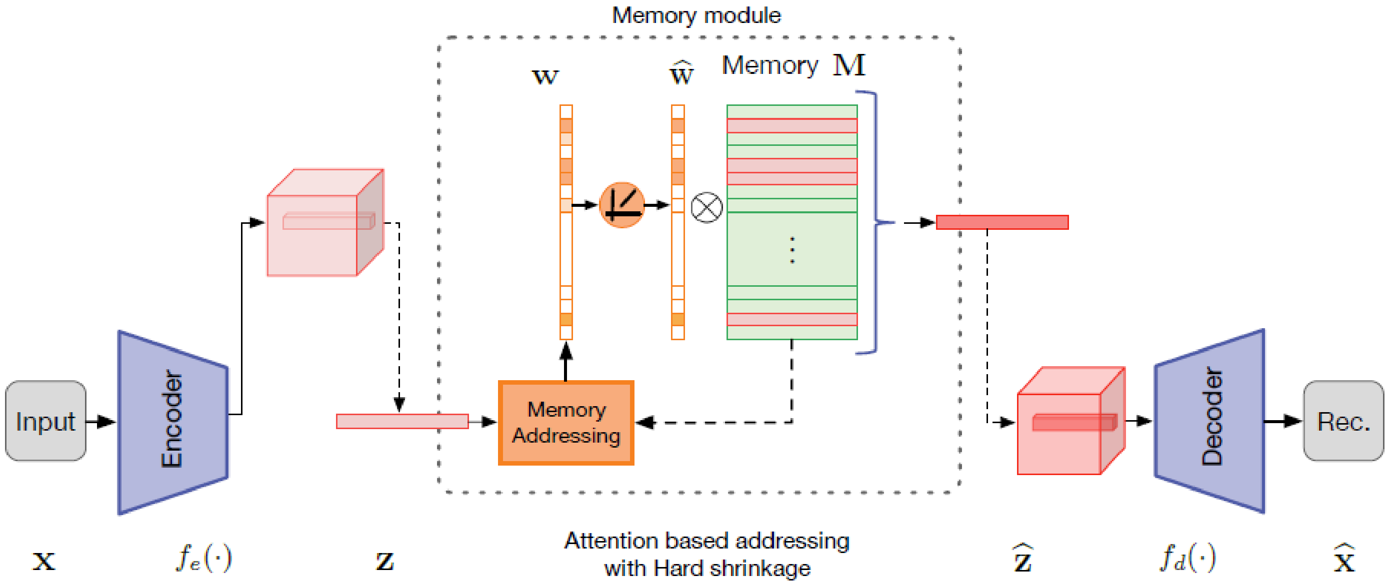
During training, the encoder and decoder are optimized to minimize the reconstruction error.

The memory contents are simultaneously updated to record the prototypical elements of the encoded normal data.

Given a testing sample, the model performs reconstruction merely using a restricted number of the normal patterns recorded in the memory.

As a result, the reconstruction tends to be close to the normal sample, resulting in small reconstruction errors for normal samples and large errors on anomalies, which will be used as a criterion to detect the anomalies

# Memorizing Normality to Detect Anomaly: MemAE

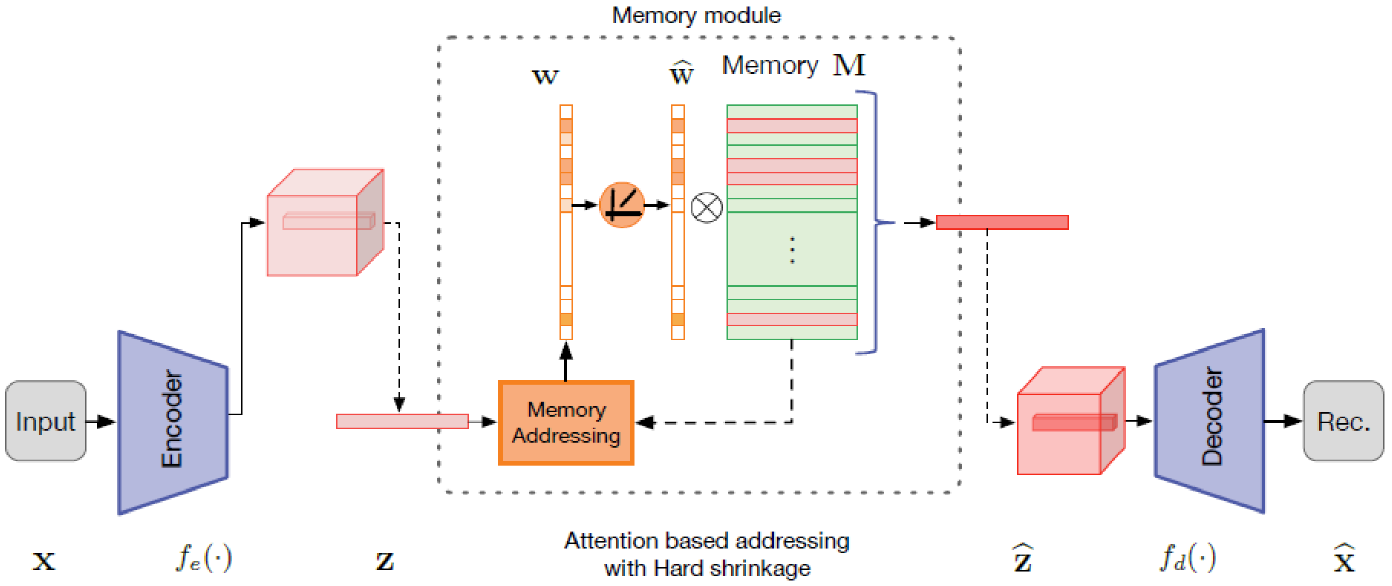



**Encoder** $f_e(\cdot) : \mathbb{X} \to \mathbb{Z}$

**Decoder** $f_d(\cdot) : \mathbb{Z} \to \mathbb{X}$

$\mathbf{z} = f_e(\mathbf{x};\ \theta_e)$ → **Memory Module** → $\hat{\mathbf{x}} = f_d(\hat{\mathbf{z}};\ \theta_d)$

# Memorizing Normality to Detect Anomaly: MemAE



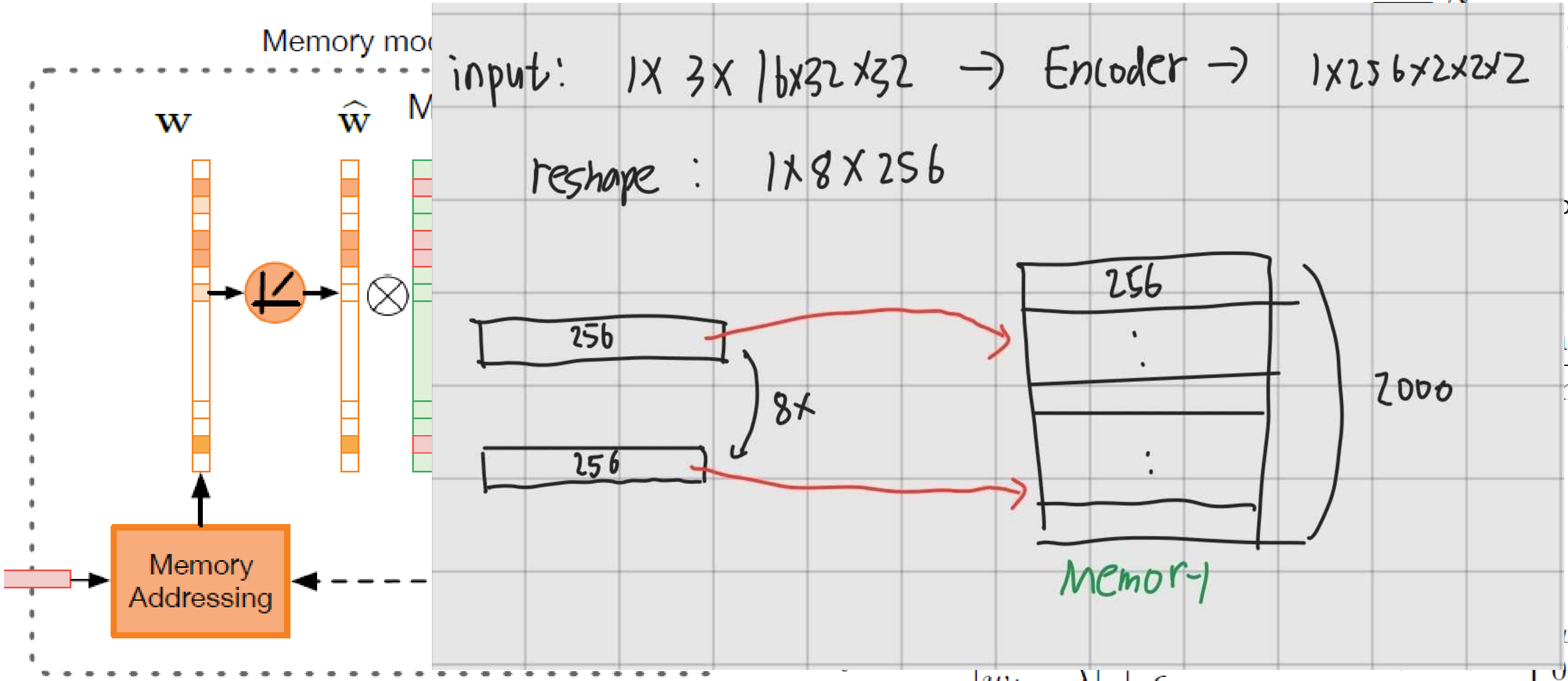

```python
from models import AutoEncoderCov3DMem
import torch

x = torch.randn(1,3,16,32,32)
model = AutoEncoderCov3DMem(chnum_in=3, mem_dim=2000)
f = model.encoder(x)    # feature
res_mem = model.mem_rep(f) # memory retrieval
f = res_mem['output']
att = res_mem['att']
output = model.decoder(f) # reconstruction
```

# Memorizing Normality to Detect Anomaly: MemAE

**Memory module**

$$\mathbb{Z} = \mathbb{R}^C$$




Attention based addressing
with Hard shrinkage

$N$

$\mathbf{m}_i,$

prototypical

$|w_i - \lambda| + \epsilon$

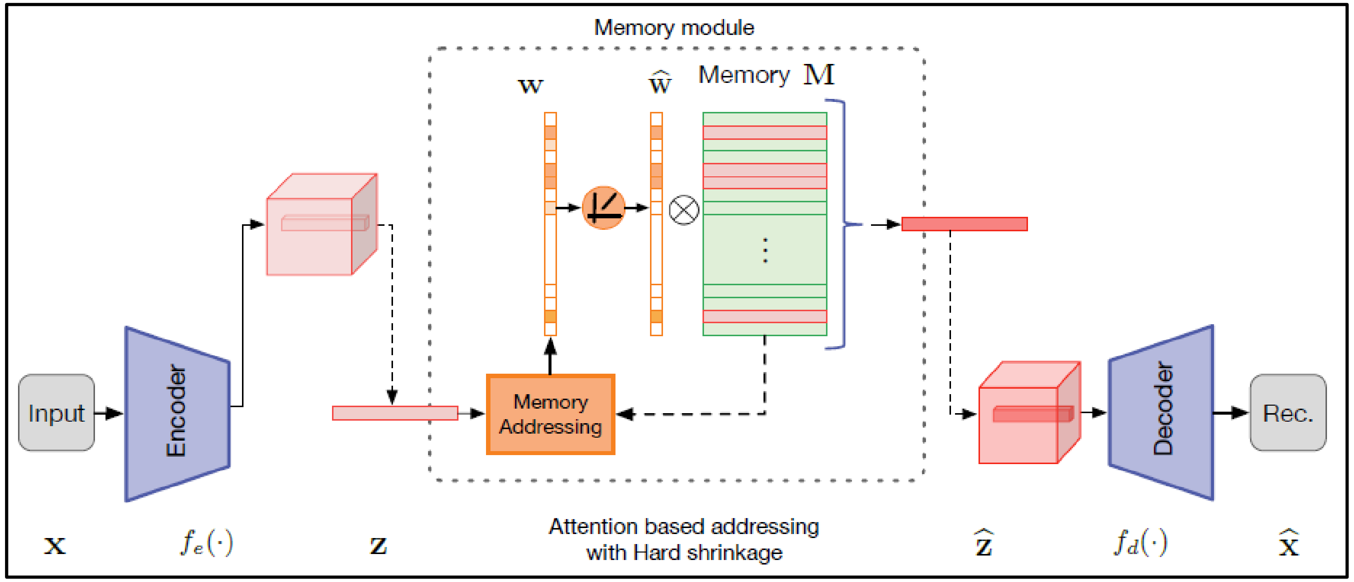
if $w_i > \lambda$,

otherwise,

After the shrinkage, we re-normalize $\widehat{\mathbf{w}}$ by letting $\widehat{w}_i = \widehat{w}_i / \|\widehat{\mathbf{w}}\|_1, \forall i$. The latent representation $\widehat{\mathbf{z}}$ will be obtained via $\widehat{\mathbf{z}} = \widehat{\mathbf{w}}\mathbf{M}$.

# Memorizing Normality to Detect Anomaly: MemAE

## Training

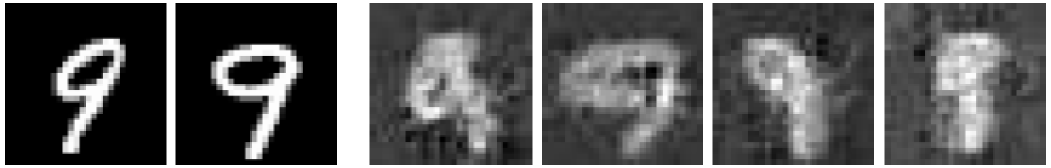



$$R(\mathbf{x}^t,\ \hat{\mathbf{x}}^t) = \|\mathbf{x}^t - \hat{\mathbf{x}}^t\|_2^2,$$

$$E(\hat{\mathbf{w}}^t) = \sum\nolimits_{i=1}^{T} -\hat{w}_i \cdot \log(\hat{w}_i).$$

$$L(\theta_e, \theta_d, \mathbf{M}) = \frac{1}{T}\sum\nolimits_{t=1}^{T} \left(R(\mathbf{x}^t,\ \hat{\mathbf{x}}^t) + \alpha E(\hat{\mathbf{w}}^t)\right)$$

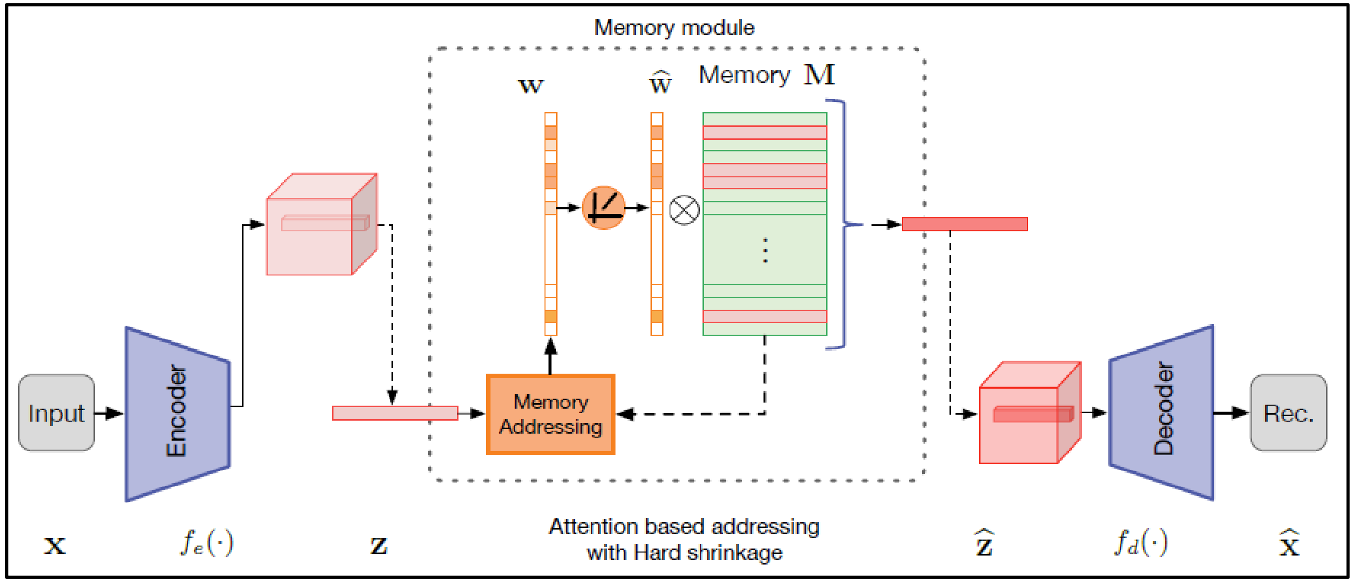
Entropy: Uniform, Gaussian 같이 고르게 퍼진 분포에서 큰값을 지닌다.
W의 entropy를 작게 학습함으로써 특정 포인트에서만 높은값을 가지는 weight vector 생성(Sparsity).
Memory module 한 포인트 포인트가 Normal 하나의 값을 크게 반영하기 위함.



(a) Training samples (b) Decoded single memory item

# Memorizing Normality to Detect Anomaly: MemAE

## 정리




**Nomarlity score:** $p_u = 1 - \frac{e_u - \min_u(e_u)}{\max_u(e_u) - \min_u(e_u)}$

$e_u$ 는 Reconstruction error

훈련:
1. 이미지 X가 인풋으로 들어오면 Encoder를 통해 z 생성, z와 M의 cosine 유사도와 Softmax 함수를 통해 w 생성
2. w에 Hard shrinkage 적용($\hat{w}$), 행렬 M과 $\hat{w}$의 곱으로 $\hat{z}$ 생성 후 Decoder를 통해 $\hat{X}$ 생성.
3. Recon + entropy Loss를 사용해 훈련

추론:
1. Encoder, Decoder, M 의 파라미터 고정. 이미지가 들어오면 z를 생성후 코사인 유사도 계산으로 w를 구함
2. w와 메모리 M의 곱으로 새로운 z 생성하고 Decoder를 통해 이미지 생성.

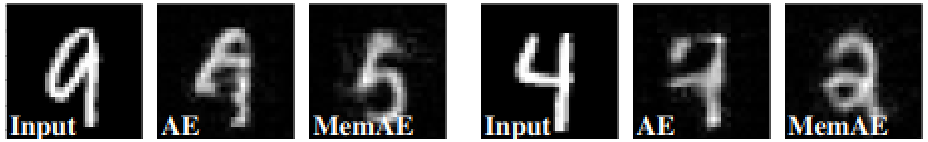
# Memorizing Normality to Detect Anomaly: MemAE

## Results

Table 1. Experimental results on image data. Average AUC values on 10 anomaly detection datasets sampled from MNIST and CIFAR-10 are shown.

| Dataset | MNIST | CIFAR-10 |
|---|---|---|
| OC-SVM [35] | 0.9499 | 0.5619 |
| KDE | 0.8116 | 0.5756 |
| VAE [18] | 0.9643 | 0.5725 |
| PixCNN [38] | 0.6141 | 0.5450 |
| DSEBM [43] | 0.9554 | 0.5725 |
| AE | 0.9619 | 0.5706 |
| MemAE-nonSpar | 0.9725 | 0.6058 |
| MemAE | **0.9751** | **0.6088** |

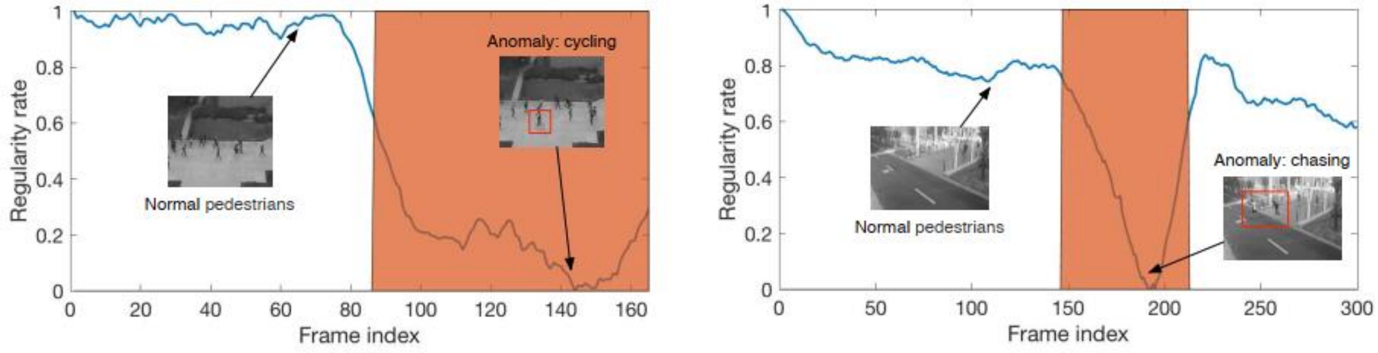
intra-class variance on several classes, which incurs unsatisfactory average ACU. Nevertheless, among the compared models with similar capacities, MemAE achieves superior performance than the competitors, which proves the effectiveness of the proposed memory module.




(a) Training on the normal "5" (b) Training on the normal "2"

Figure 4. Visualization of the reconstruction results of AE and MemAE on MNIST. (a) The models are trained on "5". The input is an image of "9". (b) The models are trained on "2". The input is an image of "4". The MemAE retrieves the normal memory items for reconstructions and obtains the results significantly different from the input anomalies.

## Results




(a) UCSD-Ped2 (b) ShanghaiTech

Figure 5. Normality scores of the video frames obtain by MemAE. The score decreases immediately when some anomalies appear in the video frame.

Table 2. AUC of different methods on video datasets UCSD-Ped2, CUHK Avenue and ShanghaiTech.

| | Method\Dataset | UCSD-Ped2 | CUHK | SH.Tech |
|---|---|---|---|---|
| Non-Recon. | MPPCA [15] | 0.693 | - | - |
| | MPPCA+SFA [27] | 0.613 | - | - |
| | MDT [27] | 0.829 | - | - |
| | AMDN [41] | 0.908 | - | - |
| | Unmasking [37] | 0.822 | 0.806 | - |
| | MT-FRCN [12] | 0.922 | - | - |
| | Frame-Pred [26] | **0.954** | **0.849** | **0.728** |
| Recon. | AE-Conv2D [11] | 0.850 | 0.800 | 0.609 |
| | AE-Conv3D [45] | 0.912 | 0.771 | - |
| | TSC [26] | 0.910 | 0.806 | 0.679 |
| | StackRNN [26] | 0.922 | 0.817 | 0.680 |
| | AE | 0.917 | 0.810 | 0.697 |
| | MemAE-nonSpar | 0.929 | 0.821 | 0.688 |
| | MemAE | **0.941** | **0.833** | **0.712** |

## Results

**Robustness to the memory size** We use the UCSD-Ped2 to study the robustness of the proposed MemAE to the memory size $N$. We conduct the experiments by using different memory size settings and show the AUC values in Figure 7. Given a large enough memory size, the MemAE can robustly produce plausible results.

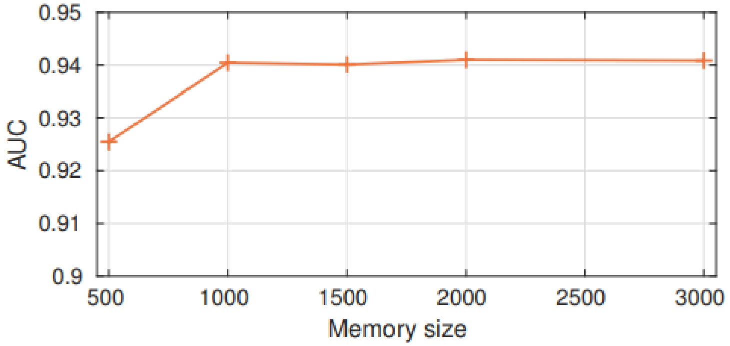



Figure 7. Robustness to the setting of memory size. AUC values of MemAE with different memory size on UCSD-Ped2 are shown.