LEARNING TO GENERATE PSEUDO ANOMALIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Due to rare occurrence of anomalous events, anomaly detection is often seen as one-class classification (OCC) problem. In this setting, an autoencoder (AE) is typically trained to reconstruct using only normal data in order to learn normalcy representations. It is expected that, at test time, the AE can well reconstruct normal data while poorly reconstructing anomalous data. However, anomalous data is often well reconstructed as well. This phenomenon can be attributed to the fact that when training AE with only normal data, the boundary between normal and abnormal data is unknown, consequently resulting in a boundary that includes the abnormal data as well. To alleviate this problem, we utilize pseudo anomalies to limit the reconstruction capability of an AE. Without imposing strong inductive bias, pseudo anomalies are generated by adding noise to the normal data. Moreover, to improve the quality of pseudo anomalies, we propose a learning mechanism to generate noise by exploiting the aforementioned weakness of AE, i.e., reconstructing anomalies too well. Evaluations on Ped2, Avenue, ShanghaiTech, and CIFAR-10 datasets demonstrate the effectiveness of our approach in improving the discriminative capability of AEs for anomaly detection.

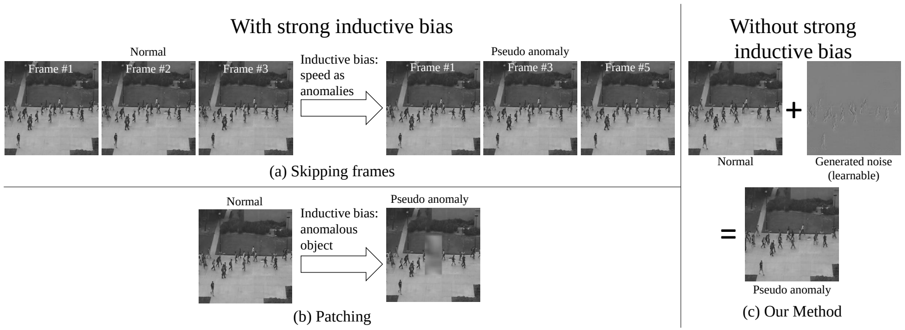
1 INTRODUCTION

Anomaly detection is one of the important components in automatic surveillance systems. Recently, it has attracted significant attention from various researchers (Liu et al., 2018a; Lee et al., 2019; Ionescu et al., 2019a; Zaheer et al., 2020a; Gong et al., 2019; Park et al., 2020; Astrid et al., 2021a; Sultani et al., 2018; Pourreza et al., 2021; Georgescu et al., 2021; Ji et al., 2020). By definition, anomalous events are rare and can be cumbersome to collect. Therefore, anomaly detection is commonly approached as one-class classification (OCC) problem in which only normal data is utilized to train a model.

Typically, an autoencoder (AE) model is utilized to tackle the OCC problem (Hasan et al., 2016; Zhao et al., 2017; Luo et al., 2017b;a; Gong et al., 2019; Park et al., 2020; Astrid et al., 2021a;b). By learning to reconstruct the normal training data, an AE encodes normalcy representations in its latent space. At test time, the model is expected to well reconstruct the normal data while poorly reconstructing the anomalies. However, as observed by Munawar et al. (2017); Gong et al. (2019); Zaheer et al. (2020a); Astrid et al. (2021a;b), AEs can oftentimes reconstruct anomalous data as well which result in a reduced capability of discrimination between normal and anomalous data.

To limit the anomalous data reconstruction capability of AEs, Gong et al. (2019); Park et al. (2020) utilize memory-based networks to memorize normal definitions in the latent space. This way, an AE is forced to reconstruct the input by using only the latent codes obtained from the learned memory. These approaches are generally successful in limiting the reconstructions of anomalous data. However, depending on the limited size of the memory, the normal data reconstructions can also be severely limited (see Fig. 6 of (Gong et al., 2019)). Therefore, it may still be difficult for such memory-augmented models to discriminate the reconstructions between normal and abnormal data.

In order for an AE to learn more suitable reconstruction boundary, recently, Astrid et al. (2021a;b) proposed the utilization of pseudo anomalies to assist the training of an AE. Pseudo anomalies are fake anomalies generated from normal data in the training set to simulate anomalous data. The AE is then trained using both normal and pseudo anomalous data. In the case of normal data, the model is trained similarly to the conventional AE, i.e., minimizing reconstruction loss. On the other hand, in the case of pseudo anomalous data, the AE is trained to not reconstruct the input, for example, by minimizing the reconstruction loss with respect to the corresponding normal data used in generating a pseudo anomaly (Astrid et al., 2021a). Although these methods outperform memory-based


With strong inductive bias
Normal
Frame #1
Frame #2
Frame #3
Inductive bias: speed as anomalies
Pseudo anomaly
Frame #1
Frame #3
Frame #5
Normal
Inductive bias: anomalous object
Pseudo anomaly
Without strong inductive bias
+
Normal
Generated noise (learnable)
=
Pseudo anomaly

(a) Skipping frames

(b) Patching

(c) Our Method

Figure 1: Comparison of pseudo anomaly generation using (a) skipping frames (Astrid et al., 2021a;b), (b) patching (Astrid et al., 2021a), and (c) our method. Our method is learnable and does not impose any strong inductive bias.

networks, one drawback of these approaches is that the pseudo anomalies are synthesized based on pre-defined assumptions, such as the speed of anomalous movements (Astrid et al., 2021a;b) (Fig. 1(a)) or anomalous objects (Astrid et al., 2021a) (Fig. 1(b)), which limits their applicability. To avoid such assumptions, we propose to construct pseudo anomalies by adding noise ΔX to the normal data (Fig. 1(c)). Moreover, we take our pseudo anomaly generation up a notch by simultaneously training an additional network that learns to generate this noise. This way, the reconstruction boundary of the AE can be improved significantly resulting in a more discriminative model.

Forcing strong inductive bias (Astrid et al., 2021a;b) to enhance learning performance is a matter of segregation among the machine learning researchers with the key argument being that having domain specific knowledge limits the idea of artificial general intelligence (Voss, 2007) and autonomous anomaly detection is certainly not an exception. Undoubtedly, inductive bias can be useful for highly specific practical anomaly detection solutions. However, this kind of methods are sensitive to the cases in which the assumptions does not apply. Meanwhile, at test time, there is no guarantee that all anomalies will conform to the assumptions. For example, in the case of fast movement as an assumption for anomalous behavior (Astrid et al., 2021a;b), the model can miss the detection of an anomalous object moving slowly. However, in certain scenarios, driving too slow may also be dangerous (Horberry et al., 2004). Moreover, a model trained in such a way may also demonstrate difficulty in detecting anomalous object with normal movement, such as an unusual object brought by a person walking normally. Using some other inductive bias, e.g., appearance of anomalous objects (Astrid et al., 2021a), is also susceptible to mis-detections of normal objects moving abnormally. Therefore, in this work, we avoid the usage of strong inductive bias in order to build a more generic method.

In summary, the contributions of our work are as follows: 1) Our work is one of the first few to explore the possibility of generating pseudo anomalies for an improved training of reconstruction models. 2) We bring the well-known weakness of AE, i.e., reconstructing anomalies too well, to our advantage by enabling our configuration to learn to generate pseudo anomalies that can hinder the successful reconstruction of anomalies. 3) Our noise-based pseudo anomaly generation assists the training without any strong inductive bias. 4) We extensively evaluate and compare our method to existing SOTA methods on highly complex video and image datasets: Ped2 (Li et al., 2013), Avenue (Lu et al., 2013), ShanghaiTech (Luo et al., 2017b), and CIFAR-10 (Krizhevsky et al., 2009).

2 Related Works

Limiting Reconstruction of AE. A common way to tackle the one-class classification (OCC) problem is by utilizing an AE to reconstruct the input (Hasan et al., 2016; Zhao et al., 2017; Luo et al., 2017b;a; Gong et al., 2019; Park et al., 2020; Astrid et al., 2021a;b). The training is carried out using only normal data with an expectation that the AE is unable to reconstruct anomalous data during test time. However, in practice, AEs can reconstruct anomalous data as well (Gong et al., 2019; Astrid et al., 2021a;b; Zaheer et al., 2020a). To alleviate the problem, Gong et al. (2019); Park et al. (2020) apply memory mechanisms to the latent space in order to limit the reconstruction capability of an AE. However, depending on the memory size, this may also limit the normal data reconstructions. Therefore, Astrid et al. (2021a;b) propose to utilize data-heuristic pseudo anomalies. To limit the reconstruction capability of the AE, the pseudo anomalies are input during training. Astrid et al. (2021b) then maximizes the reconstruction loss with respect to the pseudo anomaly input, whereas Astrid et al. (2021a) minimizes the reconstruction loss with respect to the normal data used to generate pseudo anomalies. In essence, our training configuration is similar to Astrid et al. (2021a) as we also minimize the reconstruction loss with respect to the normal data. However, we do not explicitly impose any inductive bias to generate pseudo anomalies, extending the model's anomaly detection capability to any type of possible anomalies.

Pseudo Anomalies. Several previous works take advantage of inductive bias to generate pseudo anomalies, e.g., skipping frames by assuming speed as anomalies (Astrid et al., 2021a;b) or patching by assuming the existance of anomalous objects (Astrid et al., 2021a). Our work, on the other hand, does not make any assumption by using noise-based pseudo anomalies. Moreover, we propose a method that learns to generate pseudo anomalies by adapting to the reconstruction boundary of an AE. Other works, OGNet (Zaheer et al., 2020a; 2022b) and G2D (Pourreza et al., 2021), train a binary classifier utilizing pseudo anomalies produced by an undertrained model. We discuss these methods in more details next in the adversarial training section.

Adversarial Training. In an architectural point-of-view, our method of learning to generate noise for constructing pseudo anomalies may also be seen as related to adversarial training (Goodfellow et al., 2014). However, compared to a typical adversarial architecture in which a binary classifier is used as discriminator, we utilize the AE itself as the 'discriminator'. Moreover, instead of aiming to generate real-looking data, our method attempts to generate data that is highly noisy but at the same time not impossible for the AE to reconstruct. OGNet (Zaheer et al., 2020a; 2022b) and G2D (Pourreza et al., 2021) train an AE as generator and a binary classifier as discriminator. And then, at the second phase of training, pseudo anomalies are used, which are obtained from an undertrained generator from the first phase of the training. However, such methods do not explicitly learn to generate pseudo anomalies. This potentially leaves the room for optimization. On the other hand, our method explicitly learns to generate pseudo anomalies by exploiting AE's weakness of well-reconstructing anomalies. Further discussions on the comparisons of our method with adversarial training are provided in Section 4.3.

Non-Reconstruction Methods. Different from our method, non-reconstruction methods do not utilize reconstruction as the training objective and/or as the sole decision factor of the anomaly score. There are several different approaches in this category, such as predicting future frames (Liu et al., 2018a; Park et al., 2020), utilizing object detection under the assumption that anomalous events are always related to objects (Ionescu et al., 2019a; Georgescu et al., 2021), adding optical flow components (Ji et al., 2020; Lee et al., 2019), and using a binary classifier to predict anomaly scores (Zaheer et al., 2020a; Pourreza et al., 2021).

Non-OCC Methods. To increase the discrimination capability of an AE, several researchers use real anomalies during training (Munawar et al., 2017; Yamanaka et al., 2019). Recently, video-level weakly supervised (Sultani et al., 2018; Zaheer et al., 2020b; 2022c) or fully unsupervised (Zaheer et al., 2022a) training configurations have also been introduced. Our method, on the other hand, utilizes only normal training data for training, hence not directly comparable.

With vs. Without Inductive Bias. In existing literature, several anomaly detection approaches impose inductive bias while the others do not. Object-centric based methods (Ionescu et al., 2019b; Georgescu et al., 2021) assume that every anomalous events are related to objects. However, such methods may have difficulty in detecting anomalous events not related to objects, such as unattended fires or blasts. As previously mentioned, several other pseudo anomaly based methods may require strong assumptions, such as fast movements as anomalies (Astrid et al., 2021a;b) or out-of-distribution objects as anomalies (Astrid et al., 2021a). However, such methods may fail when the anomalies do not satisfy the assumptions. We analyze such cases further in Section 4. Anomaly detection methods that do not impose inductive bias, for example, memory-based networks (Gong et al., 2019; Park et al., 2020). However, the downside is that without inductive bias, the results are below several state-of-the-art methods. To this end, in this work, with an aim to improve the performance of generic anomaly detection, we propose pseudo anomaly based method to detect anomalous events without using any inductive bias.

Denoising AE. Our work is also related to denoising AE (Salehi et al., 2021; Jewell et al., 2022; Vincent et al., 2008). However, the usage/purpose of the noise is where we are different. In a typical denoising AE, training with random noisy input is carried out to create more robust features and limit the network from duplicating the input at the output. In contrast, our noisy inputs are specifically created as pseudo anomalies. This way, the idea provides flexibility in training anomaly detection models by several ways, such as using min-max loss (Astrid et al., 2021b) or binary classifier (Zaheer et al., 2020a). That being said, noise from the previous adversarial denoising AE methods (Salehi et al., 2021; Jewell et al., 2022) can also be seen as pseudo anomalies. Since the previous works (Salehi et al., 2021; Jewell et al., 2022; Vincent et al., 2008) did not formulate the noise as pseudo anomalies, their generated noisy inputs are limited to only training denoising AE. However, with our new perspective of the noisy input as pseudo anomalies, these works can also be extended like ours, which can facilitate future research in the community.

Data Augmentation. Creating pseudo anomalies can be seen as a type of data augmentation as we add more data to the training set. However, instead of adding data of the same classes as in the typical data augmentation techniques (Bengio et al., 2011; Krizhevsky et al., 2012), creating pseudo anomalies produces new class in the training set, i.e., anomaly class. Several data augmentation techniques also utilize adversarial training to generate augmented examples that are adversarial for the model (Zhang et al., 2020; Tang et al., 2020b). Our method also utilizes adversarial training to generate pseudo anomalies. However, the definition of adversary is different. In the existing data augmentation approaches, adversary usually means the model is trained to be wrong in predicting the output or the training loss becoming higher. In our case, adversary means that the AE can well-reconstruct the generated pseudo anomalies.

Generation of Out-of-Distribution Data. Our method of generating pseudo anomalies is also related to generation of out-of-distribution data, such as Bad GAN (Dai et al., 2017), Fence GAN (Ngo et al., 2019), Margin GAN (Dong & Lin, 2019), VOS (Du et al., 2022), and BDSG (Dionelis et al., 2020). The generated out-of-distribution data is used to improve models in various applications, such as semi supervised learning (Dai et al., 2017; Dong & Lin, 2019), anomaly detection (Ngo et al., 2019), and out-of-distribution detection (Dionelis et al., 2020; Du et al., 2022). However, these methods generate low dimensional data, such as low resolution images, features, and synthetic data. Our method works on high dimensional data, i.e., sequences of higher resolution frames.

3 METHODOLOGY

In this section, our proposed approach of learning to generate pseudo anomalies is discussed. The overall configuration can be seen in Fig. 2, which mainly consists of alternate training of autoencoder $\mathcal{F}$ (Section 3.1, Fig. 2(b)) and noise generator $\mathcal{G}$ (Section 3.2, Fig. 2(a)). Details of the method are discussed next.

3.1 LEARNING NOT TO RECONSTRUCT ANOMALIES

In order to train discriminative anomaly detector model in OCC setting, in addition to the available normal training data, Astrid et al. (2021a;b) also utilize pseudo anomalies to assist the training of an autoencoder (AE). With a probability $1 - p$, the AE $\mathcal{F}$ takes a normal input X^N while taking a pseudo anomalous input X^P with a probability p:

$$\hat{X}^N = \mathcal{F}(X^N); \hat{X}^P = \mathcal{F}(X^P), \tag{1}$$

where $\hat{X}^N$ and $\hat{X}^P$ are the reconstruction outputs of the corresponding inputs.

$\mathcal{F}$ is then trained to well-reconstruct normal data and poorly-reconstruct pseudo anomalous data. In the case of normal data, similar to typical reconstruction-based models (Hasan et al., 2016; Zhao et al., 2017; Luo et al., 2017b;a; Gong et al., 2019; Park et al., 2020; Astrid et al., 2021a;b), $\mathcal{F}$ is trained to minimize the reconstruction loss:

$$\min_{\mathcal{F}} \frac{1}{T \times C \times H \times W} \left\| \hat{X}^N - X^N \right\|_F^2, \tag{2}$$

where $\|.\|_F$ means Frobenius norm.

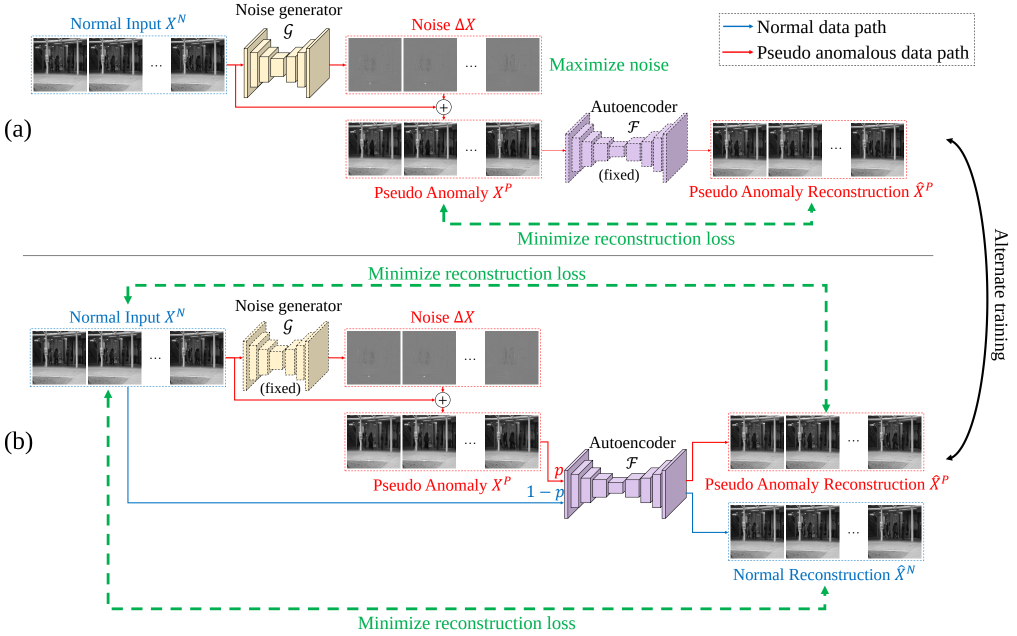

Normal Input X^N
Noise generator $\mathcal{G}$
Noise ΔX
Maximize noise
Normal data path
Pseudo anomalous data path
(a)
Autoencoder $\mathcal{F}$
(fixed)
Pseudo Anomaly X^P
Pseudo Anomaly Reconstruction $\hat{X}^P$
Minimize reconstruction loss
Alternate training
Minimize reconstruction loss
Normal Input X^N
Noise generator $\mathcal{G}$
(fixed)
Noise ΔX
(b)
Autoencoder $\mathcal{F}$
Pseudo Anomaly X^P
p
$1-p$
Pseudo Anomaly Reconstruction $\hat{X}^P$
Normal Reconstruction $\hat{X}^N$
Minimize reconstruction loss

Figure 2: Our method consists of a main autoencoder $\mathcal{F}$ and a noise generator $\mathcal{G}$ that are trained alternately: (a) A pseudo anomaly instance is constructed by adding noise to the normal data, where the noise is generated by $\mathcal{G}$. $\mathcal{G}$ learns to generate the noise by exploiting the weakness of $\mathcal{F}$, i.e., reconstructing anomalies too well. Therefore, $\mathcal{G}$ is trained to generate anomalies (maximizing noise) that are within the reconstruction boundary of $\mathcal{F}$ (minimizing reconstruction loss). (b) $\mathcal{F}$ is trained to not reconstruct anomalies by using the generated pseudo anomalies with a probability p and normal data with a probability $1-p$. During test time, only $\mathcal{F}$ is used.

Meanwhile in the case of pseudo anomalous data, we follow the setup in Astrid et al. (2021a). $\mathcal{F}$ is trained to reconstruct normal data regardless of the input, i.e., normal or pseudo anomalous. Therefore, even when the input is a pseudo anomaly, the reconstruction loss is calculated using X^N as the target:

$$\min_{\mathcal{F}} \frac{1}{T \times C \times H \times W} \left\| \hat{X}^P - X^N \right\|_F^2 . \tag{3}$$

The overall training mechanism can be seen in Fig. 2(b).

3.2 Learning to Generate Pseudo Anomalies

A pseudo anomaly is supposedly anomalous data generated from the normal training data. It is considered anomalous as it is not a part of the normalcy defined in the training data. Moreover, it is pseudo as it is not a real anomaly. In this work, we propose augmenting noise ΔX to the normal input as pseudo anomaly:

$$X^P = X^N + \Delta X. \tag{4}$$

Furthermore, we propose to train an additional AE model that learns to generate the optimal noise for a superior performance. It is pertinent to note that learning to add noise for creating pseudo anomalies does not impose any inductive bias. In order to evaluate the effectiveness of such learning approach, we also conduct experiments using a simple non-learnable Gaussian noise augmentation to generate pseudo anomalies.

3.2.1 Non-Learnable Noise.

Adding noise without any learning component can be performed by simply adding random Gaussian noise to the normal data X^N. For this purpose, the noise ΔX is defined as:

$$\Delta X = \mathcal{N}(0, \sigma), \tag{5}$$

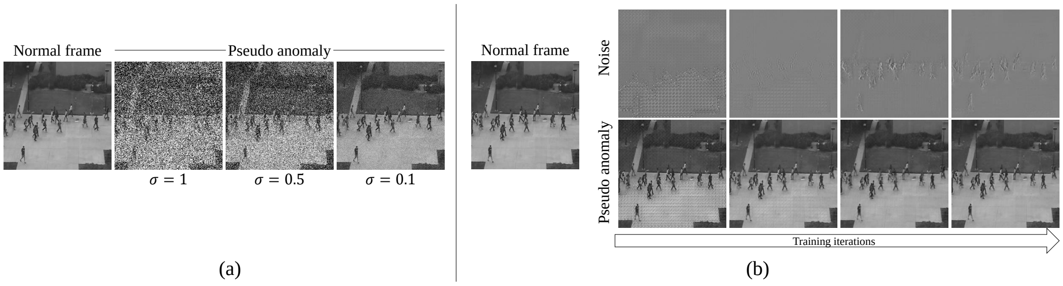


Figure 3: Visualizations of (a) pseudo anomalies constructed from a normal frame by adding Gaussian noise with various σ values, where the random noise amplitude is affected by σ; and (b) noise generated by $\mathcal{G}$ and the respective pseudo anomalies generated using our proposed *learning to generate pseudo anomalies* mechanism across different training iterations. Compared to the random noise in (a), the noise generated in our propose mechanism changes with training iterations as $\mathcal{G}$ adapts to the reconstruction boundary of $\mathcal{F}$.

where $\mathcal{N}(0, \sigma)$ is Gaussian noise with mean 0 and standard deviation σ. Several pseudo anomaly examples generated with different σ values can be seen in Fig. 3(a). $\mathcal{F}$ trained using Gaussian noise based pseudo anomalies is referred to as Gaussian noise model hereafter.

3.2.2 Learnable Noise.

In our approach, we propose to train an additional autoencoder $\mathcal{G}$ that learns to generate ΔX for creating pseudo anomalies as:

$$\Delta X = \mathcal{G}(X^N). \tag{6}$$

$\mathcal{G}$ is then trained by exploiting what we may term as the weakness of $\mathcal{F}$ in anomaly detection, i.e., reconstructing too well on anomalous data. Therefore, we propose a loss consisting of two parts, including reconstruction and noise amplitude:

$$\min_{\mathcal{G}} \frac{1}{T \times C \times H \times W} \left(\left\| \hat{X}^P - X^P \right\|_F^2 - \lambda \left\| \Delta X \right\|_F^2 \right), \tag{7}$$

where λ is a balancing hyperparameter. The first part of the training objective is to reduce the reconstruction loss of $\mathcal{F}$ for the pseudo anomalous input. Notice that, different from Equation 3, the target for the reconstruction loss in Equation 7 is X^P. In this way, $\mathcal{G}$ attempts to create noise in such a way that the resultant pseudo anomaly is within the reconstruction boundary of $\mathcal{F}$. The second part of the loss encourages the norm of the noise ΔX to be high which encourages $\mathcal{G}$ to produce high noise values. Summarily, the overall loss encourages $\mathcal{G}$ to find highly noisy pseudo anomalies that can be reconstructed by $\mathcal{F}$. Additionally, it may be noted that as $\mathcal{G}$ affects both X^P and ΔX, the backpropagation can pass through $\mathcal{G}$. The training of $\mathcal{G}$ is illustrated in Fig. 2(a).

As seen in Fig. 2, $\mathcal{G}$ is alternately trained with $\mathcal{F}$. Therefore, as the training progresses, $\mathcal{G}$ adapts to the reconstruction boundary of $\mathcal{F}$. An intuitive illustration of the overall proposed training process is visualized in Fig. 4. Moreover, Fig. 3(b) shows the evolution of noise and pseudo anomaly examples taken from Ped2 dataset across different training iterations. Interestingly, as the training proceeds, $\mathcal{G}$ learns to cover the objects of interest (i.e., pedestrians) with noise, achieving our desired target of finding noisy pseudo anomalies that can be reconstructed by $\mathcal{F}$.

4 Experiments

4.1 Datasets

We evaluate our method on several highly complex datasets including three surveillance video datasets (i.e., Ped2 (Li et al., 2013), Avenue (Lu et al., 2013), and ShanghaiTech (Luo et al., 2017b)) as well as an image dataset CIFAR-10 (Krizhevsky et al., 2009). For video datasets, the training data consists of only normal videos while every test videos contains one or more anomalous portions. For image dataset, we follow setup from Abati et al. (2019), where we set one class as normal while the others as anomaly. Since there are 10 classes, the setup is repeated for each category as normal, then the results are averaged. More details on each dataset is provided in Appendix.

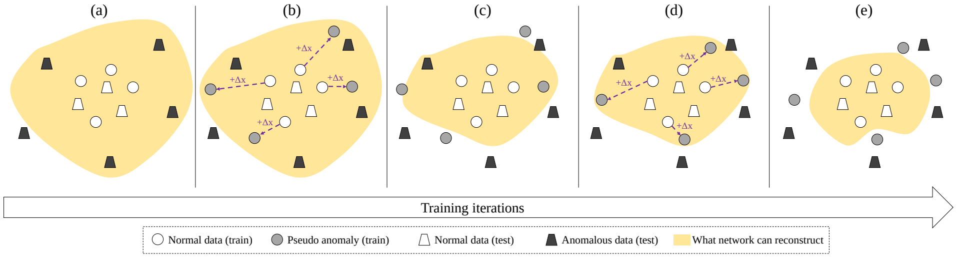


Figure 4: Illustration on how our method limits the reconstruction capability of $\mathcal{F}$ across training iterations: (a) $\mathcal{F}$ can reconstruct normal data as well as anomalous data, (b) $\mathcal{G}$ generates noise to produce pseudo anomalies within the reconstruction boundary of $\mathcal{F}$ (Equation 7, Fig. 2(a)), (c) $\mathcal{F}$ learns not to reconstruct pseudo anomalies (Equation 4, Fig. 2(b)), (d) $\mathcal{G}$ generates pseudo anomalies by adapting to the new reconstruction boundary (Equation 7, Fig. 2(a)), and (e) $\mathcal{F}$ learns not to reconstruct the new pseudo anomalies (Equation 4, Fig. 2(b)).

Methods	P	A	ST
AE-Conv2D (Hasan et al., 2016)	90.0%	70.2%	60.85%
AE-Conv3D (Zhao et al., 2017)	91.2%	71.1%	-
AE-ConvLSTM (Luo et al., 2017a)	88.10%	77.00%	-
TSC (Luo et al., 2017b)	91.03%	80.56%	67.94%
StackRNN (Luo et al., 2017b)	92.21%	81.71%	68.00%
MemAE (Gong et al., 2019)	94.1%	83.3%	71.2%
MNAD-Reconstruction (Park et al., 2020)	90.2%	82.8%	69.8%
STEAL Net (Astrid et al., 2021b)	**98.4%**	**87.1%**	73.7%
LNTRA-Patch (Astrid et al., 2021a)	94.77%	84.91%	72.46%
LNTRA-Skip frame (Astrid et al., 2021a)	96.50%	84.67%	**75.97%**
Baseline	92.49%	81.47%	71.28%
Ours-Gaussian Noise ($\sigma = 0.1$)	93.32%	81.56%	71.24%
Ours-Gaussian Noise ($\sigma = 0.5$)	93.12%	82.10%	71.73%
Ours-Gaussian Noise ($\sigma = 1$)	93.03%	82.09%	71.92%
Ours-Learnable Noise	94.57%	83.23%	73.23%

Table 1: Frame-level AUC comparisons of our approach and the existing state-of-the-art reconstruction-based methods on Ped2 (P) (Li et al., 2013), Avenue (A) (Lu et al., 2013), and ShanghaiTech (ST) (Luo et al., 2017b) datasets. Best and second best are marked as bold and underlined. Our method achieves an overall better performances compared to other generic methods, such as memory-based networks (Gong et al., 2019; Park et al., 2020). Compared to pseudo anomaly based methods that based on strong assumptions (Astrid et al., 2021a;b), our more generic method achieves a comparable performance. Comparisons with more methods are provided in Appendix.

4.2 Experimental Setup

Evaluation Criteria and Architectures. To evaluate our approach, we follow the widely popular frame-level area under the ROC curve (AUC) metric (Zaheer et al., 2020a), in which a higher AUC value represents a better performance. For $\mathcal{F}$ architecture in video and image dataset, we follow the AE in Astrid et al. (2021a) and Gong et al. (2019), respectively. We use shallower architecture for $\mathcal{G}$ by removing one layer each in encoder and decoder. More details are provided in Appendix.

Hyperparameters and Implementation Details. By default, for all of the experiments using noise-based pseudo anomalies, we set pseudo anomaly probability $p = 0.5$. Moreover, the balancing parameter λ in Equation 7 is set to 0.1 for video datasets and 5 for image dataset. In order to keep X^P consistent with the input value range of $\mathcal{F}$, i.e., $[-1, 1]$ for video datasets and $[0, 1]$ for image dataset, we clip X^P (Equation 4) into the same range. In order to incorporate the clipped values into the noise amplitude in Equation 7, ΔX is recalculated as $X^P - X^N$. The baseline hereafter in the experiments refers to $\mathcal{F}$ trained without using any pseudo anomaly, i.e., $p = 0$. See Appendix for more details.

4.3 Results on Video Data

4.3.1 Ablation Studies

In this section, we discuss the importance of the two novel components introduced in this work: adding noise to construct pseudo anomalies and the learning mechanism to generate pseudo anomalies. As seen in the last five rows of Table 1 (Reconstruction), using three benchmark datasets, we compare the baseline, our Gaussian noise model, and our learned noise model (i.e., a model trained using pseudo anomalies with the learnable noise). Compared to the baseline that does not use any pseudo anomaly for training, our Gaussian noise models trained using various values of σ generally achieve better performances. These results demonstrate the importance of noise-based

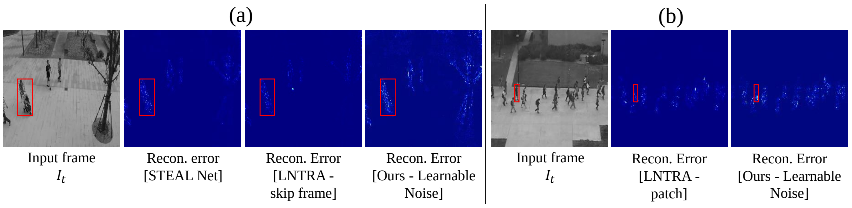


Figure 5: Reconstruction error heatmap comparisons with the other pseudo anomaly based methods (STEAL Net (Astrid et al., 2021b) and LNTRA (Astrid et al., 2021a)), which use strong inductive bias to construct pseudo anomalies, in (a) anomalous object with normal speed and (b) normal object with anomalous speed. As our method is not bounded by any inductive bias, it successfully highlights anomalous regions in the two cases where the assumptions do not hold.

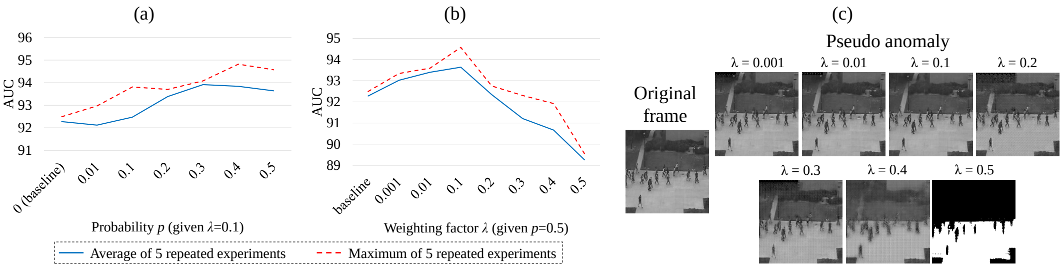


Figure 6: Hyperparameter analysis on Ped2: (a) pseudo anomaly probability p, given $\lambda = 0.1$; (b) loss weighting λ, given $p = 0.5$. Our approach is robust to a wide range of varying hyperparameters. (c) Moreover, corresponding to λ values in (b), we also provide visualizations of the generated pseudo anomalies using each λ. A sufficiently wide range of λ does not affect the performance significantly. However, too high values can lead to the generation of pseudo anomalies that are far from the normal data distribution which may not be as effective in limiting the reconstruction of AE and consequently degrade the performance.

pseudo anomalies in improving the discrimination capability of AEs for anomaly detection. Moreover, we also find that the learned noise model achieves a superior performance compared to the non-learnable Gaussian noise models. This demonstrates that the pseudo anomalies generated using the learnable noise generator help significantly in improving the anomaly detection performance. Qualitative results comparing the baseline and our model are provided in Appendix.

4.3.2 Comparisons with SOTA

In Table 1, we show the AUC comparisons of our models with the existing state-of-the-art (SOTA) approaches on Ped2 (Li et al., 2013), Avenue (Lu et al., 2013), and ShanghaiTech (Luo et al., 2017b) datasets. Due to space limitations, we present comparisons only with reconstruction-based approaches that use reconstruction of the input to detect anomalies. Comparisons with non-reconstruction methods are provided in Appendix.

Compared to the memory-based networks, overall, our learned noise model successfully outperforms MemAE (Gong et al., 2019) and MNAD-Reconstruction (Park et al., 2020). On the average taken over the individual performances of the three datasets, our model achieves 83.31% AUC, whereas MemAE and MNAD-Reconstruction reach 82.87% and 80.93% AUC, respectively. Qualitative comparisons with MemAE are provided in Appendix.

Note that, although some other approaches, such as STEAL Net (Astrid et al., 2021b) and LNTRA (Astrid et al., 2021a), achieve better performances compared to our method, they require strong inductive bias which may cause them to fail in the specific cases that deviate from the pre-defined assumptions. Comparisons of a few such possible cases are visualized in Fig. 5. Assuming abnormal speed is related to anomalous behaviors, models trained using skip frame based pseudo anomalies (Astrid et al., 2021b;a) tend to have problems in anomalies like stroller in Fig. 5(a), which is normal in its movement (i.e., normal pedestrian speed) but abnormal in its appearance (i.e., not a human). Whereas, by assuming the existence of abnormal objects, models trained using patch based pseudo

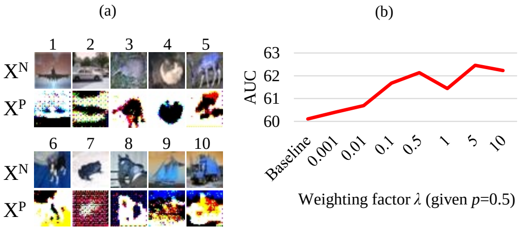


Figure 7: Results on CIFAR-10: (a) samples of normal and its corresponding pseudo anomalous data for each category; (b) AUC comparisons on different weighting factor λ values.

Method	AUC
DSEBM (Zhai et al., 2016)	57.25%
VAE (Kingma & Welling, 2013)	57.25%
MemAE (Gong et al., 2019)	60.89%
Baseline	60.10%
Ours-Gaussian Noise ($\sigma = 0.1$)	60.65%
Ours-Gaussian Noise ($\sigma = 0.5$)	60.55%
Ours-Gaussian Noise ($\sigma = 1$)	60.65%
Ours-Learnable Noise	**62.47%**

Table 2: Comparisons of our approach and other reconstruction-based SOTA methods on CIFAR-10.

anomalies (Astrid et al., 2021a) are more prone to anomalies like riding skateboard in Fig. 5(b), which is normal in its appearance (i.e., human without any noticeable abnormal object) but abnormal in its movement (i.e., faster than the normal pedestrians). In contrast, our model highlights the anomalies in both cases significantly better than the other two methods.

4.3.3 Hyperparameter Evaluations

To investigate the sensitivity of the hyperparameters in our approach, we evaluate our models on a wide range of hyperparameters. There are two hyperparameters affecting the models trained with generated noise: pseudo anomaly probability p and loss weighting parameter λ. To limit the span of experiments, we evaluate using Ped2 only. We report the average and maximum AUC out of five repeated experiments. Fig. 6(a) shows the results on different values of probability p, given a fixed $\lambda = 0.1$. The baseline is equivalent to $p = 0$. Fig. 6(b) shows the results on a wide range of λ, given $p = 0.5$. Typically, the models trained using these diverse hyperparameter values outperform the baseline which shows the robustness of our approach. However, if the noise is too high, e.g., $\lambda \geq 0.3$, the performance degrades as the pseudo anomalous data distribution may drift too far away from the normal data, hence it is not as effective in limiting the reconstruction capability of the AE. As seen in Fig. 6(c), pseudo anomalies generated using $\lambda \geq 0.3$ are very distorted from the original frame.

4.4 Results on Image Data

Quantitative results of CIFAR-10 experiments can be seen in Table 2. As seen, our learnable noise model achieves better performance compared to our baseline and Gaussian noise models. We also compare with one of the popular and well-studied SOTA method, MemAE (Gong et al., 2019). As it is a highly popular SOTA method, outperforming it places our approach as robust and effective. It may also be noted that the methods with strong assumption of anomalous behaviors in videos, such as skipping frame (Astrid et al., 2021a;b), prediction (Liu et al., 2018a; Park et al., 2020), and object-centric (Ionescu et al., 2019a; Georgescu et al., 2021), are impossible to be trained on CIFAR-10, while our approach can. This also highlights the generic applicability and superiority of our approach.

To provide with more insights, generated pseudo anomalies of CIFAR-10 are visualized in Fig. 7(a). Interestingly, unlike video datasets, strong noise works better in CIFAR-10 dataset. This may be caused by the characteristics of the anomalies, i.e., very different to the normal as they are different image categories. However, as seen in 7(b), using smaller noise also works comparably well, which shows the robustness of our method in limiting the reconstruction capability of AE also for images.

5 Conclusion

In this work, we proposed a technique to generate pseudo anomalies without imposing inductive bias by adding noise to the input. Moreover, to improve it further, we proposed to utilize an additional autoencoder that learns to generate this noise. We provided ablation studies and evaluations using Ped2, Avenue, ShanghaiTech, and CIFAR-10 datasets to demonstrate the importance of the noise addition and the training mechanism to generate noise. Even without inductive bias, our approach demonstrated superiority by achieving comparable performance to the existing state-of-the-art methods.

REFERENCES

Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 481–490, 2019.

Marcella Astrid, Muhammad Zaigham Zaheer, Jae-Yeong Lee, and Seung-Ik Lee. Learning not to reconstruct anomalies. In *British Machine Vision Conference*, 2021a.

Marcella Astrid, Muhammad Zaigham Zaheer, and Seung-Ik Lee. Synthetic temporal anomaly guided end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 207–214, October 2021b.

Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, et al. Deep learners benefit more from out-of-distribution examples. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 164–172. JMLR Workshop and Conference Proceedings, 2011.

Yunpeng Chang, Tu Zhigang, Xie Wei, and Yuan Junsong. Clustering driven deep autoencoder for video anomaly detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017.

Allison Del Giorno, J Andrew Bagnell, and Martial Hebert. A discriminative framework for anomaly detection in large videos. In *European Conference on Computer Vision*, pp. 334–349. Springer, 2016.

Nikolaos Dionelis, Mehrdad Yaghoobi, and Sotirios A Tsaftaris. Boundary of distribution support generator (bdsg): Sample generation on the boundary. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 803–807. IEEE, 2020.

Fei Dong, Yu Zhang, and Xiushan Nie. Dual discriminator generative adversarial network for video anomaly detection. *IEEE Access*, 8:88170–88176, 2020.

Jinhao Dong and Tong Lin. Margingan: Adversarial training in semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

Keval Doshi and Yasin Yilmaz. Any-shot sequential anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 934–935, 2020a.

Keval Doshi and Yasin Yilmaz. Continual learning for anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 254–255, 2020b.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Sharon Li. Towards unknown-aware learning with virtual outlier synthesis. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=TW7d65uYu5M`.

Mariana Iuliana Georgescu, Radu Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2021.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1705–1714, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 733–742, 2016.

Ryota Hinami, Tao Mei, and Shin'ichi Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3619–3627, 2017.

Tim Horberry, Laurence Hartley, Kristina Gobetti, Farlee Walker, Bankole Johnson, Steve Gersbach, and Joanne Ludlow. Speed choice by drivers: The issue of driving too slowly. *Ergonomics*, 47 (14):1561–1570, 2004.

Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7842–7851, 2019a.

Radu Tudor Ionescu, Sorina Smeureanu, Marius Popescu, and Bogdan Alexe. Detecting abnormal events in video using narrowed normality clusters. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1951–1960. IEEE, 2019b.

John Taylor Jewell, Vahid Reza Khazaie, and Yalda Mohsenzadeh. One-class learned encoder-decoder network with adversarial context masking for novelty detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3591–3601, 2022.

Xiangli Ji, Bairong Li, and Yuesheng Zhu. Tam-net: Temporal enhanced appearance-to-motion generative network for video anomaly detection. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2928. IEEE, 2009.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Sangmin Lee, Hak Gu Kim, and Yong Man Ro. Stan: Spatio-temporal adversarial networks for abnormal event detection. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 1323–1327. IEEE, 2018.

Sangmin Lee, Hak Gu Kim, and Yong Man Ro. Bman: Bidirectional multi-scale aggregation networks for abnormal event detection. *IEEE Transactions on Image Processing*, 29:2395–2408, 2019.

Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, 2013.

Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6536–6545, 2018a.

Yusha Liu, Chun-Liang Li, and Barnabás Póczos. Classifier two sample test for video anomaly detections. In *BMVC*, pp. 71, 2018b.

Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pp. 2720–2727, 2013.

Yiwei Lu, K Mahesh Kumar, Seyed shahabeddin Nabavi, and Yang Wang. Future frame prediction using convolutional vrnn for anomaly detection. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–8. IEEE, 2019.

Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*, pp. 125–141. Springer, 2020.

Weixin Luo, Wen Liu, and Shenghua Gao. Remembering history with convolutional lstm for anomaly detection. In *IEEE International Conference on Multimedia and Expo*, pp. 439–444. IEEE, 2017a.

Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 341–349, 2017b.

Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1975–1981. IEEE, 2010.

Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 935–942. IEEE, 2009.

Asim Munawar, Phongtharin Vinayavekhin, and Giovanni De Magistris. Limiting the reconstruction capability of generative neural network using negative learning. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2017.

Phuc Cuong Ngo, Amadeus Aristo Winarto, Connie Khor Li Kou, Sojeong Park, Farhan Akram, and Hwee Kuan Lee. Fence gan: Towards better anomaly detection. In *2019 IEEE 31St International Conference on tools with artificial intelligence (ICTAI)*, pp. 141–148. IEEE, 2019.

Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1273–1283, 2019.

Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14372–14381, 2020.

Masoud Pourreza, Bahram Mohammadi, Mostafa Khaki, Samir Bouindour, Hichem Snoussi, and Mohammad Sabokrou. G2d: Generate to detect anomaly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2003–2012, 2021.

Bharathkumar Ramachandra and Michael Jones. Street scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2569–2578, 2020.

Bharathkumar Ramachandra, Michael Jones, and Ranga Vatsavai. Learning a distance function with a siamese network to localize anomalies in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2598–2607, 2020.

Mohammadreza Salehi, Atrin Arya, Barbod Pajoum, Mohammad Otoofi, Amirreza Shaeiri, Mohammad Hossein Rohban, and Hamid R Rabiee. Arae: Adversarially robust training of autoencoders improves novelty detection. *Neural Networks*, 144:726–736, 2021.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6479–6488, 2018.

Che Sun, Yunde Jia, Yao Hu, and Yuwei Wu. Scene-aware context reasoning for unsupervised abnormal event detection in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 184–192, 2020.

Yao Tang, Lin Zhao, Shanshan Zhang, Chen Gong, Guangyu Li, and Jian Yang. Integrating prediction and reconstruction for anomaly detection. *Pattern Recognition Letters*, 129:123–130, 2020a.

Zhiqiang Tang, Yunhe Gao, Leonid Karlinsky, Prasanna Sattigeri, Rogerio Feris, and Dimitris Metaxas. Onlineaugment: Online data augmentation with less domain knowledge. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 313–329, Cham, 2020b. Springer International Publishing. ISBN 978-3-030-58571-6.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Peter Voss. Essentials of general intelligence: The direct path to artificial general intelligence. In *Artificial general intelligence*, pp. 131–157. Springer, 2007.

Hung Vu, Tu Dinh Nguyen, Trung Le, Wei Luo, and Dinh Phung. Robust anomaly detection in videos using multilevel representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5216–5223, 2019.

Ziming Wang, Yuexian Zou, and Zeming Zhang. Cluster attention contrast for video anomaly detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2463–2471, 2020.

Peng Wu, Jing Liu, and Fang Shen. A deep one-class neural network for anomalous event detection in complex scenes. *IEEE transactions on neural networks and learning systems*, 31(7):2609–2622, 2019.

Dan Xu, Rui Song, Xinyu Wu, Nannan Li, Wei Feng, and Huihuan Qian. Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. *Neurocomputing*, 143: 144–152, 2014.

Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156: 117–127, 2017.

Yuki Yamanaka, Tomoharu Iwata, Hiroshi Takahashi, Masanori Yamada, and Sekitoshi Kanai. Autoencoding binary classifiers for supervised anomaly detection. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 647–659. Springer, 2019.

Guang Yu, Siqi Wang, Zhiping Cai, En Zhu, Chuanfu Xu, Jianping Yin, and Marius Kloft. Cloze test helps: Effective video anomaly detection via learning to complete video events. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 583–591, 2020.

M Zaigham Zaheer, Arif Mahmood, M Haris Khan, Mattia Segu, Fisher Yu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14744–14754, 2022a.

Muhammad Zaigham Zaheer, Jin-ha Lee, Marcella Astrid, and Seung-Ik Lee. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14183–14193, 2020a.

Muhammad Zaigham Zaheer, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020b.

Muhammad Zaigham Zaheer, Jin Ha Lee, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Stabilizing adversarially learned one-class novelty detection using pseudo anomalies. *arXiv preprint arXiv:2203.13716*, 2022b.

Muhammad Zaigham Zaheer, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Clustering aided weakly supervised training to detect anomalous events in surveillance videos. *arXiv preprint arXiv:2203.13704*, 2022c.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International conference on machine learning*, pp. 1100–1109. PMLR, 2016.

Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ByxdUySKvS`.

Ying Zhang, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Shun Sakai. Video anomaly detection based on locality sensitive hashing filters. *Pattern Recognition*, 59:302–311, 2016.

Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1933–1941, 2017.

A Appendix

We provide additional experimental details and analysis in this appendix. The appendix is organized as follows:

- Datasets (A.1)
- Experimental Setup (A.2)
- Hyperparameters and Implementation Details (A.4)
  - Training Hyperparameters (A.4.1)
  - Test Time of Video Data (A.4.2)
  - Architectures (A.4.3)
- More Results on Video Data (A.5)
  - Comparisons with Non-Reconstruction Based Methods (A.5.1)
  - Training Progression (Comparisons with Adversarial Denoising AE) (A.5.2)
  - Qualitative Results on Video Data Experiments (A.5.3)

A.1 Datasets

Details of each dataset used in our experiments are as follows:

Ped2. This dataset contains 16 training and 12 test videos. The normal scenes consist of pedestrians only, whereas anomalous scenes include bikes, carts, or skateboards along with pedestrians.

Avenue. This dataset consists of 16 training and 21 test videos. Examples of anomalies are abnormal objects, such as bikes, and abnormal actions of humans, such as unusual walking directions, running, or throwing stuff.

ShanghaiTech. This is by far the largest one-class anomaly detection dataset consisting of 330 training and 107 test videos. The dataset is recorded at 13 different locations having complex lighting conditions and camera angles. In total, the test videos contain 130 anomalous events including running, riding bicycle, and fighting.

CIFAR-10. It is originally a 10 classes image classification dataset. For anomaly detection experiments, we follow setup from Abati et al. (2019), where we set one class as normal while the others as anomaly. The setup is repeated for each category as normal, then the results are averaged. We use the original split of training and test set of CIFAR-10 to train and test our model, except that we exclude the anomaly classes during training. Similar to Abati et al. (2019), we also separate 10% of the original training split for validation. The highest validated model across training epoch is evaluated.

A.2 Experimental Setup

In this section, we specify experimental setup details that have not been mentioned in the main manuscript.

A.3 Evaluation Criteria.

The ROC curve used to evaluate our method is obtained by varying the anomaly score thresholds across the whole test set, i.e., one ROC curve for a dataset.

A.4 Hyperparameters and Implementation Details.

A.4.1 Training Hyperparameters

For video dataset, we train our model using mini batch size of 4, Adam optimizer (Kingma & Ba, 2014), and learning rate of 10^{-4} for both $\mathcal{F}$ and $\mathcal{G}$. For image dataset, we set the batch size, optimizer, and learning rate to 256, Adam, and 10^{-3}, respectively.

A.4.2 Test Time of Video Data

During inference, we discard $\mathcal{G}$ and use only $\mathcal{F}$ for testing. Concurrent to Park et al. (2020); Liu et al. (2018a); Astrid et al. (2021a;b), we compute frame-level anomaly scores at test time utilizing PSNR value $\mathcal{P}_t$ between an input frame and the reconstruction:

$$\mathcal{P}_t = 10 \log_{10} \frac{M^2_{\hat{I}_t}}{\frac{1}{R} \left\| \hat{I}_t - I_t \right\|^2_F}, \tag{8}$$

where t is the frame index, I_t is the t-th frame input, $\hat{I}_t$ is the reconstruction of I_t, R is the total number of pixels in $\hat{I}_t$, and $M_{\hat{I}_t}$ is the maximum possible pixel value of $\hat{I}_t$, i.e., $M_{\hat{I}_t} = 1$.

Then, following Park et al. (2020); Liu et al. (2018a); Astrid et al. (2021a;b), min-max normalization is applied on the PSNR value to obtain the normalcy score $\mathcal{Q}_t$ of range $[0, 1]$. Finally, we calculate the anomaly score $\mathcal{A}_t$ as:

$$\mathcal{A}_t = 1 - \mathcal{Q}_t. \tag{9}$$

Following Astrid et al. (2021a); Gong et al. (2019), we calculate the anomaly score using only the 9th frame of the sequence.

A.4.3 Architectures

Video Dataset For $\mathcal{F}$, we use the same AE architecture used in Astrid et al. (2021a) with input and output values of range $[-1.0, 1.0]$. The size of each sequence is $16 \times 1 \times 256 \times 256$. The complete architecture of the AE can be seen in Table 3.

For $\mathcal{G}$, we use a shallower autoencoder consisting of three layers of Conv3D for the encoder and three layers of ConvTranspose3D as the decoder. Each of the layers, except the last ConvTranspose3D layer, is followed by a batch normalization layer and a LeakyReLU activation. The final layer of the decoder is a Tanh layer multiplied by two to generate noise of range $[-2.0, 2.0]$. The architecture can be seen in Table 4.

Image Dataset For $\mathcal{F}$, we use similar AE architecture used in Gong et al. (2019) without memory module. We add an Sigmoid layer in the end to limit the range of output to $[0, 1]$. The AE takes input of size $1 \times 3 \times 32 \times 32$. The time dimension 1 is omitted in practice. The complete architecture of the AE can be seen in Table 5.

For $\mathcal{G}$, we use shallower architecture compared to $\mathcal{F}$, as seen in Table 6. The final output is a Tanh layer to generate noise of range $[-1, 1]$.

	Layer	Output Channels	Filter Size	Stride	Padding	Negative Slope
Encoder	Conv3D	96	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	128	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
Decoder	ConvTranspose3D	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	128	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	96	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	1	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
	Tanh	-	-	-	-	-

Table 3: Architecture of $\mathcal{F}$ used in our video dataset experiments. Each number in the tuple represents time, height, and width dimensions, respectively.

	Layer	Output Channels	Filter Size	Stride	Padding	Negative Slope
Encoder	Conv3D	96	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	128	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
Decoder	ConvTranspose3D	128	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	96	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	1	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
	Tanh * 2	-	-	-	-	-

Table 4: Architecture of $\mathcal{G}$ used in our video dataset experiments. Each number in the tuple represents time, height, and width dimensions, respectively.

	Layer	Output Channels	Filter Size	Stride	Padding
Encoder	Conv2D	64	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	Conv2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	Conv2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	Conv2D	256	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
Decoder	ConvTranspose2D	256	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	ConvTranspose2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	ConvTranspose2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	ConvTranspose2D	3	4	2	0
	Sigmoid	-	-	-	-

Table 5: Architecture of $\mathcal{F}$ used in our image dataset experiments.

	Layer	Output Channels	Filter Size	Stride	Padding
Encoder	Conv2D	64	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	Conv2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	Conv2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
Decoder	ConvTranspose2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	ConvTranspose2D	128	3	2	0
	BatchNorm2D	-	-	-	-
	ReLU	-	-	-	-
	ConvTranspose2D	3	4	2	0
	Tanh	-	-	-	-

Table 6: Architecture of $\mathcal{G}$ used in our image dataset experiments.

	Methods	P	A	ST
Miscellaneous	OLED (Jewell et al., 2022)	99.02%	-	-
	STAN (Lee et al., 2018)	96.5%	87.2%	-
	MC2ST (Liu et al., 2018b)	87.5%	84.4%	-
	BMAN (Lee et al., 2019)	96.6%	**90.0%**	76.2%
	AMC (Nguyen & Meunier, 2019)	96.2%	86.9%	-
	Vu et al. (2019)	**99.21%**	71.54%	-
	DeepOC (Wu et al., 2019)	-	86.6%	-
	TAM-Net (Ji et al., 2020)	98.1%	78.3%	-
	LSA (Abati et al., 2019)	95.4%	-	72.5%
	Ramachandra et al. (2020)	94.0%	87.2%	-
	Tang et al. (2020a)	96.3%	85.1%	73.0%
	Wang et al. (2020)	-	87.0%	**79.3%**
	OGNet (Zaheer et al., 2020a)	98.1%	-	-
	Conv-VRNN (Lu et al., 2019)	96.06%	85.78%	-
	Chang et al. (2020)	96.5%	86.0%	73.3%
Object-centric	MT-FRCN (Hinami et al., 2017)	92.2%	-	-
	Ionescu et al. (2019a)	94.3%	87.4%	78.7%
	Doshi & Yilmaz (2020a;b)	97.8%	86.4%	71.62%
	Sun et al. (2020)	-	89.6%	74.7%
	VEC (Yu et al., 2020)	97.3%	89.6%	74.8%
	Georgescu et al. (2021)	**98.7%**	**92.3%**	**82.7%**
Prediction	Frame-Pred (Liu et al., 2018a)	95.4%	85.1%	72.8%
	Dong et al. (2020)	95.6%	84.9%	73.7%
	Lu et al. (2020)	96.2%	85.8%	**77.9%**
	MNAD-Prediction (Park et al., 2020)	**97.0%**	**88.5%**	70.5%
Non deep learning	MPPCA (Kim & Grauman, 2009)	69.3%	-	-
	MPPC+SFA (Kim & Grauman, 2009)	61.3%	-	-
	Mehran et al. (2009)	55.6%	-	-
	MDT (Mahadevan et al., 2010)	82.9%	-	-
	Lu et al. (2013)	-	**80.9%**	-
	AMDN (Xu et al., 2017)	90.8%	-	-
	Del Giorno et al. (2016)	-	78.3%	-
	LSHF (Zhang et al., 2016)	**91.0%**	-	-
	Xu et al. (2014)	88.2%	-	-
	Ramachandra & Jones (2020)	88.3%	72.0%	-
Reconstruction	AE-Conv2D (Hasan et al., 2016)	90.0%	70.2%	60.85%
	AE-Conv3D (Zhao et al., 2017)	91.2%	71.1%	-
	AE-ConvLSTM (Luo et al., 2017a)	88.10%	77.00%	-
	TSC (Luo et al., 2017b)	91.03%	80.56%	67.94%
	StackRNN (Luo et al., 2017b)	92.21%	81.71%	68.00%
	MemAE (Gong et al., 2019)	94.1%	83.3%	71.2%
	MNAD-Reconstruction (Park et al., 2020)	90.2%	82.8%	69.8%
	STEAL Net (Astrid et al., 2021b)	**98.4%**	**87.1%**	73.7%
	LNTRA-Patch (Astrid et al., 2021a)	94.77%	84.91%	72.46%
	LNTRA-Skip frame (Astrid et al., 2021a)	96.50%	84.67%	**75.97%**
	Baseline	92.49%	81.47%	71.28%
	Ours-Gaussian Noise ($\sigma = 0.1$)	93.32%	81.56%	71.24%
	Ours-Gaussian Noise ($\sigma = 0.5$)	93.12%	82.10%	71.73%
	Ours-Gaussian Noise ($\sigma = 1$)	93.03%	82.09%	71.92%
	Ours-Learnable Noise	94.57%	83.23%	73.23%

Table 7: Frame-level AUC comparisons of our approach and the existing state-of-the-art methods on Ped2 (P) (Li et al., 2013), Avenue (A) (Lu et al., 2013), and ShanghaiTech (ST) (Luo et al., 2017b) datasets. Best and second best in each category and dataset are marked as bold and underlined.

A.5 More Results on Video Data

A.5.1 Comparisons with Non-Reconstruction Based Methods

Table 7 shows the comparisons with state-of-the-art approaches in anomaly detection, including the non-reconstruction based methods. Following (Astrid et al., 2021a), we categorize the methods into five categories. Our method belongs to reconstruction-based methods which use reconstruction quality to measure anomaly score. Compared to non-reconstruction based methods, our method achieves a comparable performance even though we do not use assume the type of pseudo anomalies. On the other hand, the top-performers; object-centric methods require an assumption that anomalous events are always related to objects. Moreover, prediction based method predicts the future frames based on few past frames which makes the presence of anomalous movements as an assumption.

A.5.2 Training Progression (Comparisons with Adversarial Denoising AE)

Despite the architectural similarity to adversarial denoising autoencoder approaches (Salehi et al., 2021; Jewell et al., 2022), our method is rather a *cooperative* learning between the generator $\mathcal{G}$ and $\mathcal{F}$ ('discriminator'), which can be supported by the mutual loss decrease and convergence (Fig. 8 (a)). The convergence can also be seen in the AUC trend over the training epoch in Fig. 8(c). As our method can converge in a cooperative way, it is evidently more stable compared to the adversarial training methods (Salehi et al., 2021; Jewell et al., 2022) which inevitably fluctuate in the loss and therefore difficult to converge. We may also peek into the instability of Jewell et al. (2022) from its delicate selection of L1/L2 loss and hyperparameters while not providing any hyperparameter sensitivity evaluation. Moreover, both these methods only report the results on simple datasets and their performance on highly complex datasets such as ShanghaiTech is unknown.

The way $\mathcal{G}$ cooperates with $\mathcal{F}$ across the training can also be observed in Fig. 3(b). At the beginning, $\mathcal{G}$ randomly puts the noise. As $\mathcal{F}$ starts to learn to reconstruct normal, $\mathcal{G}$ then starts to generate noises around moving objects, where there are many movements so that $\mathcal{F}$ cannot easily remove the noise. But once $\mathcal{F}$ finally succeeds to remove this abnormal noises (could be regarded as abnormal patterns or behaviors), $\mathcal{G}$ has no other way but to reduce the generated noise, and this is the point where $\mathcal{F}$ and $\mathcal{G}$ agree to converge (Fig. 8(a&b)). This way, $\mathcal{G}$ helps $\mathcal{F}$ to remove abnormal patterns by presenting diverse (i.e., from high to low as seen in Fig. 8(b)) noises to $\mathcal{F}$ and $\mathcal{F}$ finally learns to reconstruct normal regardless of all these kinds of variations given from $\mathcal{G}$.

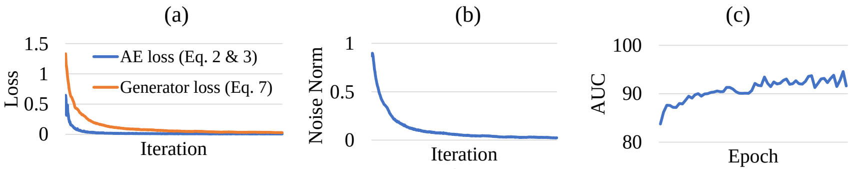


Figure 8: Training progression seen in (a) $\mathcal{F}$ and $\mathcal{G}$ loss, (b) generated noise norm, and (c) AUC, which show the stability of our approach.

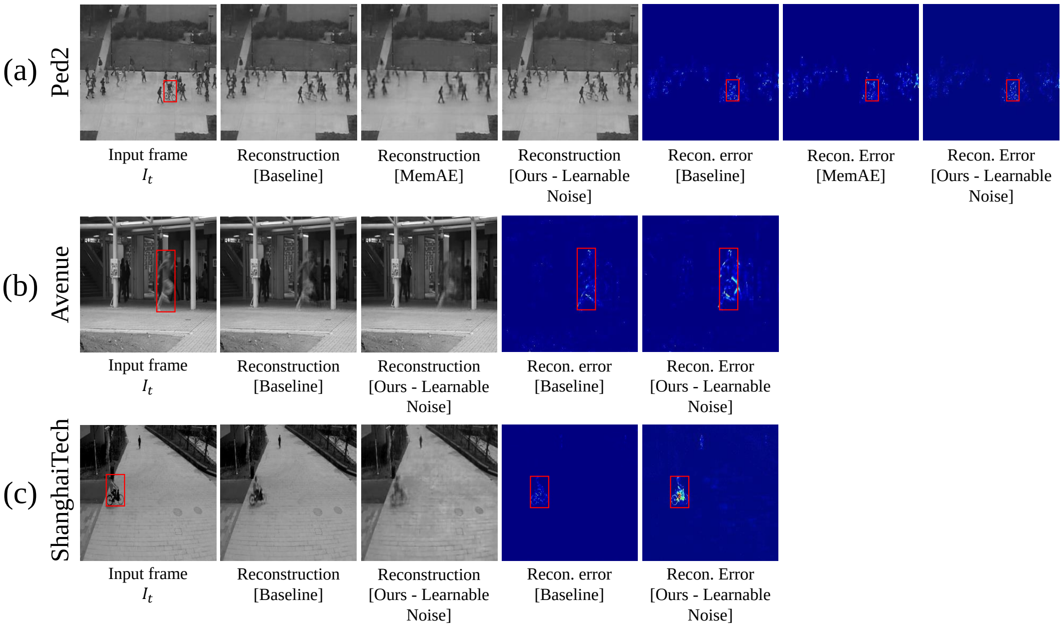


Figure 9: Input frames, reconstructions, and reconstruction error heatmaps of the baseline and our model on (a) Ped2, (b) Avenue, and (c) ShanghaiTech. Visualizations of MemAE (Gong et al., 2019) are provided only on Ped2 as the authors only made the trained model available for this dataset. As seen, our method can successfully highlight the anomalous parts (red boxes) while not producing high reconstruction errors over normal portions.

A.5.3 Qualitative Results on Video Data Experiments

To deeply understand the effects of our proposed pseudo anomalies, we qualitatively compare the baseline and our learned noise model in Fig. 9. Fig. 9 shows the input frames, reconstructions, and reconstruction heatmaps on the three benchmark datasets. The heatmaps are generated by computing the reconstruction errors followed by min-max normalization of the error values in a frame. Our method highlights the anomalous parts noticeably better than the baseline, which leads to a better discriminative capability of the model.

We also provide qualitative results comparison of our method with MemAE on Ped2 in Fig. 9(a). Although MemAE successfully distorts the anomalous regions, we can observe that it also distorts the normal regions which reduces its capability to discriminate between normal and abnormal. Our method, on the other hand, retains the reconstruction quality of the normal regions while distorting the anomalous regions. This leads to the superior performance of our model compared to the memory-based networks.