Learning Normal Dynamics in Videos with Meta Prototype Network

Hui Lv[1], Chen Chen[2], Zhen Cui[1]*, Chunyan Xu[1], Yong Li[1], Jian Yang[1]
[2]PCALab, Nanjing University of Science and Technology, [2]University of North Carolina at Charlotte
{hubrthui, zhen.cui, cyx, yong.li, csjyang}@njust.edu.cn, chen.chen@uncc.edu

Abstract

Frame reconstruction (current or future frame) based on Auto-Encoder (AE) is a popular method for video anomaly detection. With models trained on the normal data, the reconstruction errors of anomalous scenes are usually much larger than those of normal ones. Previous methods introduced the memory bank into AE, for encoding diverse normal patterns across the training videos. However, they are memory-consuming and cannot cope with unseen new scenarios in the testing data. In this work, we propose a dynamic prototype unit (DPU) to encode the normal dynamics as prototypes in real time, free from extra memory cost. In addition, we introduce meta-learning to our DPU to form a novel few-shot normalcy learner, namely Meta-Prototype Unit (MPU). It enables the fast adaption capability on new scenes by only consuming a few iterations of update. Extensive experiments are conducted on various benchmarks. The superior performance over the state-of-the-art demonstrates the effectiveness of our method. ***Our code is available at*** https://github.com/ktr-hubrt/MPN/.

1. Introduction

Video anomaly detection (VAD) refers to the identification of behaviors or appearance patterns that do not conform to the expectation [2, 3, 5, 28]. Recently, there is a growing interest in this research topic because its key role in surveillance for public safety, *e.g.* the task of monitoring video in airports, at border crossings, or at government facilities becomes increasingly critical. However, the 'anomaly' is conceptually unbounded and often ambiguous, making it infeasible to gather data of all kinds of possible anomalies. Anomaly detection is thus typically formulated as an unsupervised learning problem, aiming at learning a model to exploit the regular patterns only with the normal data. During inference, patterns that do not agree with the encoded regular ones are considered as anomalies.

*Corresponding authors

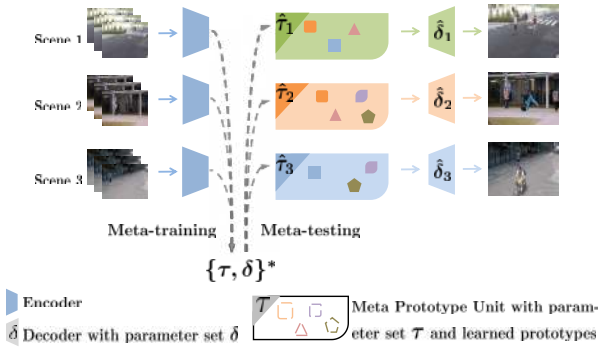


Figure 1: An overview of our approach. (1) We design a Dynamic Prototype Unit (DPU) to learn a pool of prototypes for encoding normal dynamics; (2) Meta-learning methodology is introduced to formulate the DPU as a few-shot normalcy learner – Meta Prototype Unit (MPU). It improves the scene adaption capacity by learning an initialization of the target model and adjusting it to new scenes with parameters update during inference. Better viewed in color.

Deep Auto-Encoder (AE) [38] is a popular approach for video anomaly detection. Researchers usually adopt AEs to model the normal patterns with historical frames and to reconstruct the current frame [11, 31, 39, 4, 40, 1] or predict the upcoming frame [22, 34, 24, 26, 10]. For simplicity, we refer to the two cases as frame prediction. Since the models are trained with only normal data, higher prediction errors are expected for abnormal (unseen patterns) inputs than those of the normal counterparts. Previously, many methods are based on this assumption for anomaly detection. However, this assumption does not always hold true.

On the one hand, the existing methods rely on large volumes of normal training data to model the shared normal patterns. These models are prone to face the 'over-generalizing' dilemma, where all video frames can be predicted well, no matter they are normal or abnormal, owing to the powerful representation capacity of convolutional neural networks (CNNs) [37, 10]. Previous approaches [37, 10] proposed to explicitly model the shared normal patterns across normal training videos with a memory bank, for

the propose of boosting the prediction of normal regions in frames while suppressing the abnormal ones. However, it is extremely memory-consuming for storing the normal patterns as memory items across the whole training set.

To tackle this limitation, we propose to encode the normal dynamics in an attention manner, which is proven to be effective in representation learning and enhancement [46, 20, 13]. A normalcy learner, named as Dynamic Prototype Unit (DPU), is developed to be easily incorporated into the AE backbone. It takes the encoding of consecutive normal frames as input, then learns to mine diverse normal dynamics as compact prototypes. More specifically, we apply a novel attention operation on the AE encoding map, which assigns a normalcy weight to each pixel location to form a normalcy map. Then, prototypes are obtained as an ensemble of the local encoding vectors under the guidance of normalcy weights. Multiple parallel attention operations are applied to generate a pool of prototypes. With the proposed compactness and diverseness feature reconstruction loss function, the prototype items are trained to represent diverse and compact dynamics of the shared normal patterns in an end-to-end fashion. Finally, the AE encoding map is aggregated with the normalcy encoding reconstructed by prototypes for latter frame prediction.

On the other hand, the normal patterns appearing in various scenes differ from each other. For instance, a person running in a walking zone is regarded as an anomaly, while this activity is normal in the playground. Previous methods [22, 10] assume the normal patterns in training videos are consistent with those of test scenes in the unsupervised setting of VAD. However, this assumption is unreliable, especially in real-world applications where surveillance cameras are installed in various places with significantly different scenarios. Therefore, there is a pressing need to develop an anomaly detector with adaption capability. To this end, [37] defines a rule for updating items in the memory bank based on a threshold to record normal patterns and ignore abnormal ones. However, it is impossible to find a uniform and optimal threshold for distinguishing the normal and abnormal frames under various scenarios.

In this work, we approach this problem from a new perspective, motivated by [25], which is the few-shot setting for video anomaly detection. In the few-shot setting, videos from multiple scenes are accessible during training, and a few video frames from target scene are available during inference. A solution to this problem is using the meta-learning technique. In this meta-training phase, a few-shot target model is trained to adapt to a new scene with a few frames and parameters update iterations. The procedure is repeated using video data from different scenes for obtaining a model initialization that serves as a good starting point for fast adaption to new scenes. Therefore, we formulate our DPU module as a few-shot normalcy learner, namely Meta Prototype Unit (MPU), for the goal of learning to learn the normalcy in target scenes. Rather than roughly shifting to the new scene by adjusting the whole network [25], which may lead to the 'over-generalizing' problem, we propose to freeze the pre-trained AE and only update the parameters of our MPU. Consuming only a few parameters and update iterations, our meta-learning model is endued with the power of fast and effective adaption to the normalcy of unseen scenarios. An overview of our approach is presented in Fig. 1.

We summarize our contributions as follows: i) We develop a Dynamic Prototype Unit (DPU) for learning to represent diverse and dynamic patterns of the normal data as prototypes. An attention operation is thus designed for aggregating the normal dynamics to form prototype items. The whole process is differentiable and trained end-to-end. ii) We introduce meta-learning into our DPU and improve it as a few-shot normalcy learner – Meta Prototype Unit (MPU). It effectively endows the model with the fast adaption capability by consuming only a few parameters and update iterations. iii) Our DPU-based AE achieves new state-of-the-art (SOTA) performance on various unsupervised anomaly detection benchmarks. In addition, experimental results validate the adaption capability of our MPU in the few-shot setting.

2. Related Work

Anomaly Detection. Due to the absence of anomaly data and expensive costs of annotations, video anomaly detection has been formulated into several types of learning problems. For example, the unsupervised setting assumes only normal training data [19, 27, 23], and weakly-supervised setting can access videos with video-level labels [43, 53, 28]. In this work, we focus on the unsupervised setting, which is more practical in real applications. For example, the normal video data of surveillance cameras are easily accessible for learning models describing the normality. Earlier methods, based on sparse coding [7, 51, 23], markov random field [14], a mixture of dynamic textures [30], a mixture of probabilistic PCA models [15], *etc*., tackle the task as a novelty detection problem [28]. Latter, deep learning (CNNs in particular) has triumphed over many computer vision tasks including video anomaly detection (VAD). In [27], Luo *et al*. propose a temporally coherent sparse coding-based method which can be mapped to a stacked RNN framework.

Recently, many methods leverage deep Auto-Encoder (AE) to model regular patterns and reconstruct video frames [11, 31, 39, 4, 40, 1]. Multiple variants of AE have been developed to cooperate spatial and temporal information for video anomaly detection. In [26, 6], the authors investigate Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) for modeling regular patterns in se-

quential data. Liu *et al*. [22] propose to predict the future frame with AE and Generative Adversarial Network (GAN). They assume anomalous frames are unpredictable in the video sequence. It has achieved superior performance over previous reconstruction-based methods. However, this kind of methodology suffers from the 'over-generalizing' problem that sometimes anomalous frames can also be predicted well (*i.e*. small prediction error) as normal ones.

Gong *et al*. (MemAE) [10] and Park *et al*. (LMN) [37] introduce a memory bank into the AE for anomaly detection. They record normal patterns across training videos as memory items in a bank, which brings extra memory cost. While we propose to learn the normalcy with an attention mechanism to measure the normal extent. The learning procedure is fully differentiable and the prototypes are dynamically learned with the benefits of adapting to the current scene spatially and temporally, compared with querying and updating the memory bank with pre-defined rules for recording rough patterns cross the training data in [10, 37]. Moreover, the prototypes are automatically derived based on the real-time video data during inference, without referencing to the memory items collected from the training phase [10, 37]. For adaption to test scenes, Park *et al*. [37] further expand the update rules of the memory bank by using a threshold to distinguish abnormal frames and record normal patterns. However, it is impossible to find a uniform and optimal threshold for distinguishing the normal and abnormal frames under various scenarios. On the contrary, we introduce the meta-learning technology into our DPU module to enable the fast adaption capacity to a new scenery.

Attention Mechanisms. Attention mechanism [48, 49, 13, 42, 16, 50, 9, 52, 20] is widely adopted in many computer vision tasks. Current methods can be roughly divided into two categories, which are the channel-wise attention [49, 13, 42, 50] and spatial-wise attention [42, 52, 49, 16, 9]. SENet [13] designs an effective and lightweight gating mechanism to self-recalibrate the feature map via channel-wise importance. Wang *et al*. [48] propose a trunk-and-mask attention between intermediate stages of a CNN. However, most prior attention modules focus on optimizing the backbone for feature learning and enhancement. We propose to leverage the attention mechanism to measure the normalcy of spatial local encoding vectors, and use them to generate prototype items which encode the normal patterns.

Few-Shot and Meta-learning. In few-shot learning, researchers aim to mimic the fast and nimble learning ability of humans, which can quickly adapt to a new scenario with only a few data examples [18]. Generally, meta-learning has been developed to tackle this problem. The meta-learning methods mainly fall into three categories: metric-based [17, 47, 44], model-based [41, 33] and optimization-based approaches [8]. These methods can quickly adapt to a new task through the meta-update scheme among multiple tasks during parameter optimization. However, most of the approaches above are designed for simple tasks like image classification. Recently, Lu *et al*. [25] follow the optimization-based meta-learning approach [8] and apply it to train a model for scene-adaptive anomaly detection. They simply set the whole network as the few-shot target model for meta-learning, for learning an initialization parameter set of the entire model. However, in this work, we learn two sets of initial parameters and update step sizes separately for an elaborate updating of designed module in our model with fewer parameters and update iterations.

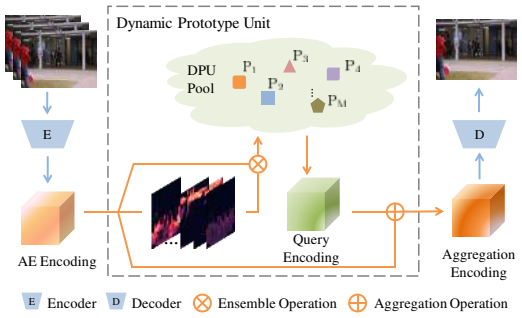


Figure 2: The framework of DPU-based model. The proposed Dynamic Prototype Unit (DPU) is plugged into an Auto-Encoder (AE) to learn prototypes for encoding normal dynamics. The prototypes are obtained from the AE encoding with the guidance of normalcy weights and the normalcy weights of the AE encoding are generated in a fully differentiable attention manner. Then an normalcy encoding map (green color) is reconstructed as an encoding of learned prototypes. It is further aggregated with the AE encoding map for latter frame prediction.

3. Method

In this section, we elaborate the proposed method for VAD. First, we describe the learning process of normal dynamics in the Dynamic Prototype Unit (DPU) in Sec. 3.1, and we explain the objective functions of the framework in Sec. 3.2. Then in Sec. 3.3, we present the details of the few-shot normalcy learner. Finally in Sec. 3.4, we detail the training and testing procedures of our VAD framework.

3.1. Dynamic Prototype Unit

The framework of the DPU-based AE is shown in Fig. 2. DPU is trained to learn and compress normal dynamics of real-time sequential information as multiple prototypes and enrich the input AE encoding with normal dynamics information. Note that, DPU can be plugged into different places (with different resolutions) of the AE. We conduct ablation studies in Sec. 4.4 to analyze the impact of DPU position.

Let's first consider an AE model takes as inputs the T observed video frames $(\mathrm{I}_{\mathrm{k-T+1}}, \mathrm{I}_{\mathrm{k-T+2}}, ..., \mathrm{I}_{\mathrm{k}})$, simplified

as x_k. Then the selected hidden encoding of AE is feed forward into our DPU $P_\tau : \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times\mathrm{c}} \rightarrow \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times\mathrm{c}}$. Finally, the output encoding of DPU is run through the remaining AE layers (after DPU) for predicting upcoming ground-truth frame $y_k = \mathrm{I}_{\mathrm{k}+1}$. We denote the frame sequence as an input&output pair (x_k, y_k) of the k-th moment.

The forward pass of DPU is realized by generating a pool of dynamic prototypes in a fully differentiable attention manner, then reconstructing a normalcy encoding by retrieving the prototypes, and eventually aggregating the input encoding with the normalcy encoding as the output. The whole process can be broken down into 3 sub-processes, which are $Attention$, $Ensemble$ and $Retrieving$.

Concretely, the t-th input encoding map $\mathcal{X}_\mathrm{t} = f_\theta(x_\mathrm{t}) \in \mathbb{R}^{\mathrm{h,w,c}}$ from AE is first extracted, viewed as $\mathrm{N} = \mathrm{w} * \mathrm{h}$ vectors of c dimensional, $\{\mathrm{x}_\mathrm{t}^1, \mathrm{x}_\mathrm{t}^2, ..., \mathrm{x}_\mathrm{t}^\mathrm{N}\}$. In the sub-process of $Attention$, a quantity of M attention mapping functions $\{\psi_m : \mathbb{R}^\mathrm{c} \rightarrow \mathbb{R}^1\}_{m=1}^M$ are employed to assign normalcy weights to encoding vectors, $w_\mathrm{t}^{n,m} \in \mathcal{W}_\mathrm{t}^m = \psi_m(\mathcal{X}_\mathrm{t})$. On each pixel location, the normalcy weight measures the normalcy extent of the encoding vector. Here, $\mathcal{W}_\mathrm{t}^m \in \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times 1}$ denotes the m-th normalcy map, generated from the m-th attention function. Then one unique prototype p_t^m is derived as an ensemble of N encoding vectors with normalized normalcy weights in sub-process $Ensemble$ as:

$$\mathrm{p}_\mathrm{t}^m = \sum_{n=1}^{\mathrm{N}} \frac{w_\mathrm{t}^{n,m}}{\sum_{n'=1}^{\mathrm{N}} w_\mathrm{t}^{n',m}} \mathrm{x}_\mathrm{t}^n. \tag{1}$$

Similarly, M prototypes are derived from multiple attention functions to form a prototype pool, $\mathcal{P}_\mathrm{t} = \{\mathrm{p}_\mathrm{t}^m\}_{m=1}^M$.

Finally, in the $Retrieving$ sub-process, input encoding vectors x_t^n ($n \in \mathrm{N}$) from the AE encoding map are used as queries to retrieve relevant items in the prototype pool for reconstructing a normalcy encoding $\widetilde{\mathrm{X}}_\mathrm{t} \in \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times\mathrm{c}}$. For every obtained normalcy encoding vector, this proceeds as:

$$\tilde{\mathrm{x}}_\mathrm{t}^n = \sum_{m=1}^{M} \beta_\mathrm{t}^{n,m} \mathrm{p}_\mathrm{t}^m, \tag{2}$$

where $\beta_\mathrm{t}^{n,m} = \frac{\mathrm{x}_\mathrm{t}^n \mathrm{p}_\mathrm{t}^m}{\sum_{m'=1}^M \mathrm{x}_\mathrm{t}^n \mathrm{p}_\mathrm{t}^{m'}}$ denotes the relevant score between the n-th encoding vector x_t^n and the m-th prototype item p_t^m. The obtained normalcy map is aggregated with the original encoding $\mathcal{X}$ as the final output using a channel-wise sum operation. The key idea is to enrich the AE encoding with the normalcy information to boost the prediction of normal parts of video frames while suppressing the abnormal parts. The output encoding of DPU goes through the remaining AE layers for later frame prediction.

3.2. VAD Objective Functions

In this section, we present the objective functions in the pipeline, which enable the prototype learning for normalcy dynamics representation, feature reconstruction for normalcy enhanced encoding, and frame prediction for anomaly detection. To train our model, the overall loss function $\mathcal{L}$ consists of a feature reconstruction term $\mathcal{L}_\mathrm{fea}$ and a frame prediction term $\mathcal{L}_\mathrm{fra}$. These two terms are balanced by weight λ_1 as:

$$\mathcal{L} = \mathcal{L}_\mathrm{fra} + \lambda_1 \mathcal{L}_\mathrm{fea}. \tag{3}$$

Frame Prediction Loss is formulated as the L2 distance between ground-truth y_t and network prediction $\hat{y}_\mathrm{t}$:

$$\mathcal{L}_\mathrm{fra} = \|\hat{y}_\mathrm{t} - y_\mathrm{t}\|_2. \tag{4}$$

Feature Reconstruction Loss is designed to make the learned normal prototypes have the properties of compactness and diversity. It has two terms $\mathcal{L}_\mathrm{c}$ and $\mathcal{L}_\mathrm{d}$, aiming at the two properties respectively, and is written as:

$$\mathcal{L}_\mathrm{fea} = \mathcal{L}_\mathrm{c} + \lambda_2 \mathcal{L}_\mathrm{d}, \tag{5}$$

where λ_2 is the weight parameter. The compactness term $\mathcal{L}_\mathrm{c}$ is for reconstruction of normalcy encoding with compact prototypes. It measures the mean L2 distance of input encoding vectors and their most-relevant prototypes as:

$$\mathcal{L}_\mathrm{c} = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \|\mathrm{x}_\mathrm{t}^n - \mathrm{p}_\mathrm{t}^*\|_2, \tag{6}$$

$$\mathrm{s.t.}, * = \underset{m\in[1,M]}{\mathrm{argmax}}\, \beta_\mathrm{t}^{n,m}, \tag{7}$$

where $\beta^{n,m}$ is the relevant score mentioned in Eq. 2. Note that, argmax is only used to obtain indices of the most relevant vector, and not involved in the back-propagation. We further promote the diversity among prototype items by pushing the learned prototypes away from each other. The diversity term $\mathcal{L}_\mathrm{d}$ is expressed as:

$$\mathcal{L}_\mathrm{d} = \frac{2}{M(M-1)} \sum_{m=1}^{M} \sum_{m'=1}^{M} [-||p_m - p_{m'}||_2 + \gamma]_+. \tag{8}$$

Here, γ controls the desired margin between prototypes. Taking benefits of above two terms, the prototype items are encouraged to encode compact and diverse normalcy dynamics for normal frame prediction.

3.3. Meta-learning in Few-shot VAD

Generally, the AEs take consecutive video frames as inputs and reconstruct the current frame or predict the subsequent frame. In this work, we focus on the latter paradigm. We first consider a VAD architecture formulated as $f_\theta(E_\eta(x)) = D_\delta(P_\tau(E_\eta(x)))$, where η, δ denote the parameters of the AE encoding/decoding function E, D, respectively. The designed model takes as input a sequence of frame samples x. Then the AE encoding $\mathcal{X} = E_\eta(x)$ is fed

into the DPU module P_τ. DPU learns to encode the normal dynamics information in consecutive video frames with the parameter set τ. Our few-shot target model $f_\theta(\mathcal{X})$, namely Meta-Prototype Unit (MPU), consists of the main module DPU and the AE decoder with parameter set $\theta = \tau \cup \delta$. Taking the subsequent frame sample y as the ground-truth, the target model is updated based on the objective functions defined in Sec. 3.2. The process is denoted as the update function U with frame pair (x, y).

During inference, short normal clips of test videos are available for adjusting the model to the new scenery in the few-shot setting of VAD. To mimic this adaption process, meta-training strategy is implemented in the training phase. In meta-training, a good initialization θ_0 is pursued so that the target model, starting from θ_0 and applying one or a few iterations of update function U, can quickly adapt to a new scenery with limited data samples. We adopt the gradient-descent style update function [21, 36] which is parameterized by α. Then the function U is formulated as:

$$U(\theta, \nabla_\theta \mathcal{L}; \alpha) = \theta - \alpha \odot \nabla_\theta \mathcal{L}. \tag{9}$$

$\mathcal{L}$ is the designed loss function (Eq. 3) for the target model. $\odot$ denotes the element-wise product. α is the parameter that controls the step size of one update iteration, and it is set to the same size as parameter set θ.

To ensure the robustness of scene adaption, during meta-training, the target model is updated and supervised based on the error signals from different input&output pairs in one scene. The key idea is that the target model should also generalize to other frames in the same scene, not only several frames which the model is trained on. Given a random input&output pair (x_k, y_k) from a normal video, one update step of the target model with initialization θ_0 is derived as:

$$\theta_0^{i+1} = U(\theta_0^i, \nabla_{\theta_0^i} \mathcal{L}(y_k, f_{\theta_0}(E_\eta(x_k)))). \tag{10}$$

After T update iterations, scene-adapted model parameters $\hat{\theta}$ are obtained. We denote the round of T update iterations as an episode. The iterations number T in an episode is set to 1, to guarantee a fast adaption capability. Then we evaluate the model with $\hat{\theta}$ to minimize the scene error signal by running the network through a randomly sampled input&output pair (x_j, y_j) in the same scene as (x_k, y_k).

The gradients of function of gradients algorithm [32, 29, 8, 36] is applied to compute the gradients of above objective function for obtaining a good initialization model θ_0^* and update step size α^* as:

$$\theta_0^*, \alpha^* = \underset{\theta_0, \alpha}{\operatorname{argmin}} \mathbb{E}[\mathcal{L}(y_j, f_{\hat{\theta}}(E_\eta(x_j)))]. \tag{11}$$

3.4. Video Anomaly Detection Pipeline

We first explain the details of the whole network architecture and how anomaly scores are generated. Then we describe the training and testing phases of our framework.

Network Architecture Details. Our framework is implemented as a single end-to-end network illustrated in Fig. 2. We adopt the same network architecture in [22, 37] as the backbone of AE to facilitate a fair comparison. In the DPU module, M attention mapping functions are implemented as fully connected layers to generate a series of normalcy maps and further to form a pool of dynamic prototypes. The output encoding of DPU is put forward through the decoder of AE for frame prediction. In addition, the DPU module is meta-trained as a few-shot learner, *i.e.* Meta Prototype Unit (MPU). The details are explained below.

Anomaly Score. To better quantify the anomalous extent of a video frame during inference, we investigate the two cues of feature reconstruction and frame prediction. Since the normal dynamics items in the dynamic prototype pool are learned to encode the compact representations of the normal encoding as in Eq. 5, during inference, an anomaly score can be naturally obtained by measuring the compactness error of feature reconstruction term as: $\mathcal{S}_{\text{fea}} = \mathcal{L}_{\text{c}}(\mathcal{X}_{\text{t}}, \mathcal{P}_{\text{t}})$. $\mathcal{X}_{\text{t}}$ and $\mathcal{P}_{\text{t}}$ denote the input encoding map and the dynamic prototype pool of the t-th moment, respectively. As in previous methods [22, 10, 37], frame prediction error is also leveraged as an anomaly descriptor: $\mathcal{S}_{\text{fra}} = \mathcal{L}_{\text{fra}}(\hat{y}_{\text{t}}, y_{\text{t}})$. Thus we obtain above two kinds of anomaly scores and combine them with a balance weight λ_{s} as: $\mathcal{S} = \mathcal{S}_{\text{fra}} + \lambda_{\text{s}} \mathcal{S}_{\text{fea}}$.

Training Phase. Before meta-training, the AE backbone is first pre-trained using only frame prediction loss (Eq. 4). Then, in a meta-training episode, we randomly sample K tuples of double input&output pairs $\{[(x_i, y_i), (x_j, y_j)]_{i \neq j}\}_{k=1}^{K}$ from a video – K-shot, for parameter update in Eq. 10 and signal backward in Eq. 11. Multiple episodes with K-shot data sampled from different videos are constructed as a training mini-batch. After several times of training epochs with frame pairs sampled from videos of diverse scenes, an initialization parameter set θ_0^* is obtained, ready for scene adaption.

Testing Phase. In the testing phase, given a new test sequence, we simply use the first several frames of the sequence to construct K-shot input&output frame pairs for updating model parameters. The same procedure is used in the meta-training phase. The updated model is used for detecting anomalies afterwards.

4. Experiments

4.1. Problem Settings, Datasets and Setups

Problem Settings. For better evaluating the effectiveness of our approach, we follow two anomaly detection problem settings, which are the unsupervised setting and few-shot setting. The first one is widely adopted in existing literature [37, 10, 22, 19, 23, 27], where only normal videos are available during training. The trained models are used to

detect anomalies in test videos. Note that the scenarios of test videos are seen during training in this setting. The second one, for meta-learning evaluation, is based on collecting training and testing videos from different datasets to make sure the diversity of scenarios during training and testing. This setting is also called 'cross-dataset' testing in [25]. In summary, the first setting challenges the approaches for how well they can perform under one fixed camera, while the latter setting examines the adaption capability, when given a new camera. We believe above settings are essential for evaluating a robust and practical anomaly detection method.

Datasets. Four popular anomaly detection datasets are selected to evaluate our approach under different problem settings. 1) The UCSD Ped1 & Ped2 dataset [19] contains 34 and 16 training videos, 36 and 12 test videos, respectively, with 12 irregular events, including riding a bike and driving a vehicle. 2) The CUHK Avenue dataset [23] consists of 16 training and 21 test videos with 47 abnormal events such as running and throwing stuff. 3) The ShanghaiTech dataset [27] contains 330 training and 107 test videos of 13 scenes. 4) The UCF-Crime dataset [43] contains normal and crime videos collected from a large number of real-world surveillance cameras where each video comes from a different scene. We use the 950 normal videos from this dataset for meta-training, then test the model on other datasets in the cross-dataset testing as in [25].

Evaluation Metrics. Following prior works [22, 26, 30], we evaluate the performance using the area under ROC curve (AUC). ROC curve is obtained by varying the threshold for the anomaly score for each frame-wise prediction.

Implementation Details. Input frames are resized to the resolution of 256×256 and normalized to the range of $[-1, 1]$. During the AE pre-training, the model is trained with the learning rate as 0.0001 and batch size as 4. In the default setting, DPU is plugged into the AE after the third CNN layer counting backwards, with the encoding feature map of resolution $256 \times 256 \times 128$. Training epochs are set to $60, 60, 60, 10$ on Ped1, Ped2, Avenue and Shanghai Tech, respectively. During meta training, the AE backbone is frozen, only the few-shot target model MPU is trained. The learning rate of the update iteration of the MPU parameter set θ is set to 0.00001 for 1000 training epochs. The mini-batch is set as 10 episodes, and the learning rate of step size α is 0.00001. The balance weights in the objective functions are set as $\lambda_1 = 1$, $\lambda_2 = 0.01$. The desired margin γ in feature diversity term is set to 1. Finally, the hyper-parameter λ_s is set to 1. The experiments are conducted with four Nvidia RTX-2080Ti GPUs.

4.2. Comparisons with SOTA Methods

Evaluation under the unsupervised setting. We first perform an experiment to show that our proposed backbone architecture is comparable to the state-of-the-arts. Note that this sanity check uses the standard training/test setup (training set and testing set are provided by the original datasets), and our model can be directly compared with other existing methods. Table 1 shows the comparisons among our proposed architecture and other methods when using the standard unsupervised anomaly detection setup on several anomaly detection datasets. MemAE [10] and LMN [37] are most-related methods to our approach. They learn a large memory bank for storing normal patterns across the training videos. While we propose to learn a few dynamic normal prototypes conditioned on input data, which is more memory-efficient. The superior performance also demonstrates the effectiveness of our DPU module. On ped1 and Shanghai Tech, AUCs of our approach are lower than those of rGAN [25]. This is reasonable because the model architecture of rGAN is more complicated. rGAN uses a ConvLSTM to retain historical information by stacking AE several times. However, we only apply a single AE.

Table 1: Quantitative comparison with state-of-the-art methods for anomaly detection. We measure the average AUC (%) on UCSD Ped1 & Ped2 [19], CUHK Avenue [23], and ShanghaiTech [27] in the unsupervised setting. Numbers in bold indicate the best performance and underscored ones are the second best.

Methods	Ped1	Ped2	Avenue	Shanghai
MPPCA [14]	59.0	69.3	-	-
MPPC+SFA [14]	68.8	61.3	-	-
MDT [30]	81.8	82.9	-	-
MT-FRCN [12]	-	92.2	-	-
Unmasking [45]	68.4	82.2	80.6	-
SDOR [35]	71.7	83.2	-	-
ConvAE [11]	75.0	85.0	80.0	60.9
TSC [27]	-	91.0	80.6	67.9
StackRNN [27]	-	92.2	81.7	68.0
Frame-Pred [22]	83.1	95.4	85.1	72.8
AMC [34]	-	96.2	86.9	-
rGAN* [25]	83.7	95.9	85.3	73.7
rGAN [25]	**86.3**	96.2	85.8	**77.9**
MemAE [10]	-	94.1	83.3	71.2
LMN [37]	-	**97.0**	<u>88.5</u>	70.5
Ours w/o DPU.	83.2	95.1	84.0	66.7
Ours w DPU.	<u>85.1</u>	<u>96.9</u>	**89.5**	<u>73.8</u>

Evaluation under the few-shot setting. To demonstrate the scene adaption capacity of our approach, we conduct cross-dataset testing by meta-training on the training set of Shanghai Tech and normal videos of UCF-Crime, and then using the other datasets (UCSD Ped1, UCSD Ped2, CUHK Avenue) for validation. The comparison results are reported in Table 2. As we can see, on most circumstances, the pre-trained DPU model is more generalizing than rGAN. Feature reconstruction based on prototypes largely boosts the robustness of anomaly detection with frame prediction. Furthermore, $4 \sim 5\%$ gain can be achieved with our MPU (10-shot to 0-shot) on various benchmarks. The performance of our MPU-based AE is superior/comparable to the SOTA few-shot learner (rGAN (Meta)) [25], with a significantly

Table 2: Comparison of K-shot ($K = 0, 1, 5, 10$) scene-adaptive anomaly detection under the cross-dataset testing setting. Note that $K = 0$ represents the models are only pre-trained without any adaption.

Shanghai Tech

Target	Methods	0-shot (K=0)	1-shot (K=1)	5-shot (K=5)	10-shot (K=10)
UCSD Ped 1	rGAN [25] (Finetune)	73.1	76.99	77.85	78.23
	rGAN [25] (Meta)	73.1	**80.6**	**81.42**	**82.38**
	Ours (Meta)	**74.45**	78.54	79.35	80.20
UCSD Ped 2	rGAN [25] (Finetune)	81.95	85.64	89.66	91.11
	rGAN [25] (Meta)	81.95	91.19	91.8	92.8
	Ours (Meta)	**90.17**	**94.46**	**94.67**	**95.75**
CUHK Avenue	rGAN [25] (Finetune)	71.43	75.43	76.52	77.77
	rGAN [25] (Meta)	71.43	76.58	77.1	78.79
	Ours (Meta)	**74.06**	**78.92**	**80.25**	**81.69**

UCF crime

Target	Methods	0-shot (K=0)	1-shot (K=1)	5-shot (K=5)	10-shot (K=10)
UCSD Ped 1	rGAN [25] (Finetune)	66.87	71.7	74.52	74.68
	rGAN [25] (Meta)	66.87	**78.44**	**81.43**	**81.62**
	Ours (Meta)	**75.52**	77.19	78.33	79.53
UCSD Ped 2	rGAN [25] (Finetune)	62.53	65.58	72.63	78.32
	rGAN [25] (Meta)	62.53	83.08	86.41	**90.21**
	Ours (Meta)	**86.04**	**88.43**	**87.83**	89.89
CUHK Avenue	rGAN [25] (Finetune)	64.32	66.7	67.12	70.61
	rGAN [25] (Meta)	64.32	72.62	74.68	79.02
	Ours (Meta)	**82.26**	**85.62**	**85.66**	**85.91**

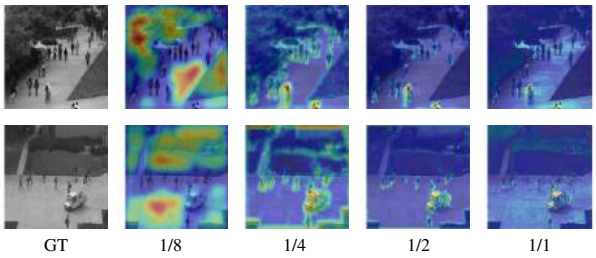


Figure 3: Visualization of AE encoding activation maps from the perspective of L2_norm. GT stands for ground-truth frame and the annotations of other columns denote the corresponding ratios of input images resolution (256×256).

faster adaption and inference speed. We provide more detailed model complexity and inference speed in Sec. 4.3.

4.3. Model Complexity and Inference Speed

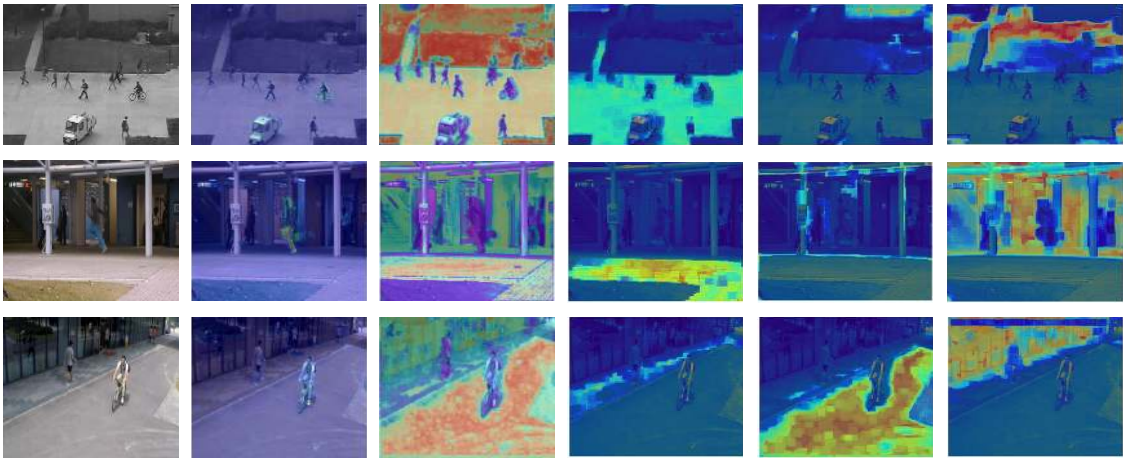
With a single Nvidia RTX-2080Ti GPU, our model can run at 166.8 FPS. Note that our DPU module only consumes 1.28K extra parameters (with 10 prototypes). Although the parameter size of MemAE [10] is smaller than that of ours, the large memory bank used in MemAE leads to a time-consuming read operation, so as the whole inference procedure. Apart from model parameters, our model does not need extra memory space for prototypes, which can be viewed as latent feature vectors. Moreover, the inference speed of our method is almost 80 $\times$ faster than rGAN [25]. The update iteration for scene adaption of our model ($K = 1$) takes only 0.04 seconds (23.9 FPS). This is almost 19 $\times$ faster than rGAN [25] ($K = 1$) which takes 0.75 seconds (1.3 FPS). The fast inference speed makes our method more favorable in real-world applications.

Table 3: Analysis on the model complexity and inference speed of various SOTA methods. The inference speed information is collected by running the official implements on a single Nvidia RTX-2080Ti GPU on a machine with 4 CPU cores of E5-2650 v4@2.20GHz and 27.5 G memory.

Methods	Parameters (M)	FPS
rGAN [25]	19.0	2.1
MemAE [10]	6.2	86.7
LMN [37]	15.0	126.3
Ours	12.7	166.8

4.4. Ablation Studies

Model Component Analysis. We first analyze the effectiveness of DPU. We set $M = 10$ as the default number of the attention mapping functions in DPU. The results are listed in Table 4. It is clear that the overall performances on various benchmarks are boosted with our DPU by a large margin. We also visualize some example prediction error maps as well as the normalcy maps in DPU in Fig. 4. To better analyze the learned normalcy maps, we aggregate all



(a) (b) (c) (d) (e) (f)

Figure 4: Visualization of some examples of test cases and DPU normalcy maps. The groups of pictures in different columns denote (a) ground-truth frame, (b) error map, (c) sum of normalcy map in DPU, (d) $\sim$ (f) various normalcy map, respectively.

Table 4: AUC analysis of the designed DPU module. In the table, FR and FP stand for anomaly scores derived from the Feature Reconstruction and the Frame Prediction, respectively.

Setting	Shanghai	Avenue	Ped2	Ped1
AE baseline (FP)	66.7	83.9	95.1	83.2
AE with DPU (FP)	71.1	85.2	92.6	83.5
AE with DPU (FR)	71.9	87.1	96.2	74.1
AE with DPU (FP & FR)	**73.8**	**89.5**	**96.9**	**85.1**

Table 5: Analysis on the plugging spot of the DPU module. The resolution is divided by the resolution of input images (256×256).

Resolution	**1/1**	1/2	1/4	1/8
AUC	**89.19**	86.72	84.66	81.18

M maps by the sum operation as in Fig. 4 (c). The normalcy maps encode diverse normal attributes of the scenes such as roads, grasses, and buildings, shown in the columns of (d) $\sim$ (f). Furthermore, the weights in suspicious regions are far smaller than those in other parts of the map, indicating that the normal patterns are well encoded as prototypes.

DPU Resolution Analysis. To investigate the effect of the plugging spot of the DPU module, we carry out experiments on four positions with encoding maps of different resolutions. The results are listed in Table 5. The AUC results are derived from the feature reconstruction anomaly score on Ped2 dataset. The performance increases along with the resolution. We visualize the activation map of the encoding using the L2-norm of the spatial encoding vectors in Fig. 3. The higher the activation value, the more information is included in the encoding vector. We find that in the higher resolution layers of AE, more anomaly cues are included, which is beneficial for measuring the anomalous extent with the feature reconstruction.

Prototype Quantity Analysis. To encode the normal dynamics as prototypes, we propose to leverage multiple attention mapping functions for measuring the normalcy of encoding vectors and deriving prototypes as ensembles of the vectors. The number of the attention functions, also denoting the quantity of prototypes, serves as the up-bound of the diverse prototypes needed in one scenario. Experimental results on Ped2 are in Table 6. Based on the results, $M = 10$ is an appropriate number of required prototypes. With the number increasing, more noise information is involved and the diversity of prototype items can not be guaranteed, leading to a drastic decline of the performance.

Table 6: AUC analysis on the quantity of prototypes in DPU.

Number	1	5	**10**	20	40
FR	87.69	90.49	**92.59**	88.26	84.37
FP	94.86	95.45	**96.22**	95.70	95.11
Overall	95.22	95.57	**96.90**	96.03	95.74

5. Conclusion

In this work, we have introduced a prototype learning module to explicitly model the normal dynamics in video sequences with an attention mechanism for unsupervised anomaly detection. The prototype module is fully differentiable and trained in an end-to-end manner. Without extra memory consumption, our approach achieves SOTA performance on various anomaly detection benchmarks in the unsupervised setting. In addition, we improve the prototype module as a few-shot normalcy learner with the meta-learning technology. Extensive experimental evaluations demonstrate the efficiency of the scene-adaption approach.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grants Nos. 62072244, 61972204, 61906094), the Natural Science Foundation of Jiangsu Province (Grant No. BK20190019).

References

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *CVPR*, 2019. 1, 2

[2] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *TPAMI*, 2008. 1

[3] Yannick Benezeth, P-M Jodoin, Venkatesh Saligrama, and Christophe Rosenberger. Abnormal events detection based on spatio-temporal co-occurences. In *CVPR*, 2009. 1

[4] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017. 1, 2

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 2009. 1

[6] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *International Symposium on Neural Networks*, 2017. 2

[7] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR*, 2011. 2

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3, 5

[9] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 3

[10] 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 *ICCV*, 2019. 1, 2, 3, 5, 6, 7

[11] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *CVPR*, 2016. 1, 2, 6

[12] Ryota Hinami, Tao Mei, and Shin'ichi Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *CVPR*, 2017. 6

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 2, 3

[14] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009. 2, 6

[15] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009. 2

[16] Idan Kligvasser, Tamar Rott Shaham, and Tomer Michaeli. xunit: Learning a spatial activation function for efficient image restoration. In *CVPR*, 2018. 3

[17] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, 2015. 3

[18] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015. 3

[19] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *TPAMI*, 2013. 2, 5, 6

[20] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, 2019. 2, 3

[21] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *ArXiv*, 2017. 5

[22] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *CVPR*, 2018. 1, 2, 3, 5, 6

[23] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, 2013. 2, 5, 6

[24] Yiwei Lu, Mahesh Kumar Krishna Reddy, Seyed shahabeddin Nabavi, and Yang Wang. Future frame prediction using convolutional vrnn for anomaly detection. In *AVSS*, 2019. 1

[25] Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *ECCV*, 2020. 2, 3, 6, 7

[26] Weixin Luo, Wen Liu, and Shenghua Gao. Remembering history with convolutional lstm for anomaly detection. In *ICME*, 2017. 1, 2, 6

[27] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *ICCV*, 2017. 2, 5, 6

[28] Hui Lv, Chuanwei Zhou, Chunyan Xu, Zhen Cui, and Jian Yang. Localizing anomalies from weakly-labeled videos. *ArXiv*, 2020. 1, 2

[29] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *ICML*, 2015. 5

[30] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010. 2, 6

[31] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, 2011. 1, 2

[32] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017. 5

[33] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, 2017. 3

[34] Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *ICCV*, 2019. 1, 6

[35] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *CVPR*, 2020. 6

[36] Eunbyung Park and Alexander C Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *ECCV*, 2018. 5

[37] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *CVPR*, 2020. 1, 2, 3, 5, 6, 7

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[39] Mohammad Sabokrou, Mahmood Fathy, and Mojtaba Hoseini. Video anomaly detection and localization based on the sparsity and reconstruction error of auto-encoder. *Electronics Letters*, 2016. 1, 2

[40] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *CVPR*, 2018. 1, 2

[41] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 3

[42] Kai Su, Dongdong Yu, Zhenqi Xu, Xin Geng, and Changhu Wang. Multi-person pose estimation with enhanced channel-wise and spatial information. In *CVPR*, 2019. 3

[43] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, 2018. 2, 6

[44] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 3

[45] Radu Tudor Ionescu, Sorina Smeureanu, Bogdan Alexe, and Marius Popescu. Unmasking the abnormal events in video. In *ICCV*, 2017. 6

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[47] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, 2016. 3

[48] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, 2017. 3

[49] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 3

[50] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, 2018. 3

[51] Bin Zhao, Li Fei-Fei, and Eric P Xing. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR*, 2011. 2

[52] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018. 3

[53] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *CVPR*, 2019. 2