Lightning Fast Video Anomaly Detection via Multi-Scale Adversarial Distillation

Florinel-Alin Croitoru, Nicolae-Cătălin Ristea, Dana Dăscălescu, Radu Tudor Ionescu, Fahad Shahbaz Khan, Mubarak Shah

\textsuperscript{a}Department of Computer Science, University of Bucharest, 14 Academiei Street, Bucharest 010014, Romania
\textsuperscript{b}National University for Science and Technology Politehnica Bucharest, 313 Splaiul Independentei Street, Bucharest, Romania
\textsuperscript{c}Securifi, 21D Mirea Voda, Bucharest 030662, Romania
\textsuperscript{d}MBZ University of Artificial Intelligence, Masdar City, Abu Dhabi, UAE
\textsuperscript{e}Linköping University, 581 83 Linköping, Sweden
\textsuperscript{f}Center for Research in Computer Vision (CRCV), University of Central Florida, Orlando 32816, FL, US

ABSTRACT

We propose a very fast frame-level model for anomaly detection in video, which learns to detect anomalies by distilling knowledge from multiple highly accurate object-level teacher models. To improve the fidelity of our student, we distill the low-resolution anomaly maps of the teachers by jointly applying standard and adversarial distillation, introducing an adversarial discriminator for each teacher to distinguish between target and generated anomaly maps. We conduct experiments on three benchmarks (Avenue, ShanghaiTech, UCSD Ped2), showing that our method is over 7 times faster than the fastest competing method, and between 28 and 62 times faster than object-centric models, while obtaining comparable results to recent methods. Our evaluation also indicates that our model achieves the best trade-off between speed and accuracy, due to its previously unheard-of speed of 1480 FPS. In addition, we carry out a comprehensive ablation study to justify our architectural design choices. Our code is freely available at: \url{https://github.com/ristea/fast-aed}

1. Introduction

Video anomaly detection is a difficult task, being actively studied in recent years [Accetino et al., 2022, Doshi and Yilmaz, 2020, Georgescu et al., 2021, Gong et al., 2019, Ionescu et al., 2019b, Li et al., 2022, Liu et al., 2021, Lu et al., 2020, Nguyen and Meunier, 2019, Pang et al., 2019, Ramachandra et al., 2022, Ristea et al., 2022, Smeureanu et al., 2017, Sun et al., 2020, Tang et al., 2020, Vu et al., 2019, Wang et al., 2020, Yu et al., 2020, 2022b, Zaheer et al., 2022] due to its increasing importance. The difficulty of the task stems from the fact that abnormal events depend on the context and occur rarely. To illustrate the scarcity and reliance on context of abnormal events, we consider the truck deliberately driven into the Berlin Christmas market\textsuperscript{1} in 2016 as a relevant example. A truck driven on the road is labeled as a normal event, but, as soon as the truck intrudes into a pedestrian area, e.g. a market, the event becomes abnormal. This example confirms that labeling an event as normal or abnormal depends on the context. Moreover, according to Wikipedia\textsuperscript{2}, there are about 170 vehicle ramming attacks registered since 1953 to date, but many of them are too old to be caught on video. Hence, our example also confirms that such events rarely occur.

Due to the rarity and reliance on context of abnormal events, it is nearly impossible to collect a sufficiently representative set of such events, eliminating the possibility of applying traditional approaches based on supervised learning to perform video anomaly detection. Without the supervised option at hand, researchers turned their attention to alternative solutions, the most prominent alternative being based on outlier detection [Cong et al., 2011, Dutta and Banerjee, 2015, Hasan et al., 2016, Ionescu et al., 2019b, Kim and Grauman, 2009, Lee et al., 2019, Liu et al., 2018a, Lu et al., 2013, Luo et al., 2017, Park et al., 2020, Ramachandra and Jones, 2020, Ramachandra et al., 2020, 2022, Sabokrou et al., 2017, Wu et al., 2019, Zhong et al., 2019]. Methods based on outlier detection learn

\textsuperscript{1}https://en.wikipedia.org/wiki/2016_Berlin_truck_attack
\textsuperscript{2}https://en.wikipedia.org/wiki/Vehicle-ramming_attack
a normality model from training videos containing only normal events, labeling test samples that deviate from the model as abnormal. Our approach falls within the same line of research, being based on distilling knowledge from object-centric outlier detection systems (Georgescu et al., 2021, 2022b), which reported state-of-the-art performance levels in anomaly detection. We hereby emphasize that top scoring abnormal event detection systems such as (Barbălău et al., 2023; Georgescu et al., 2021, 2022b; Ionescu et al., 2019a; Liu et al., 2021) operate at the object level, relying on a pre-trained object detector which limits the real-time processing to only a single video stream per GPU. Although it might seem sufficient to process one video stream at about 20-30 FPS on one GPU, the cost of computational resources to process tens or hundreds of video streams (as expected in real-world surveillance scenarios) quickly becomes very high, considering that the price range of one GPU is around $1000-4000. The high cost of the computational resources adds to the difficulty of solving the video anomaly detection task in the real-world.

To address the aforementioned efficiency problem, we propose a fast anomaly detection model based on a novel convolutional transformer that replaces fully connected layers inside the transformer with pointwise convolutional layers. Our model operates at the frame level, eliminating the need for a pre-trained object detector. At the same time, we aim to reach an accuracy level that is as close as possible to that of object-centric models. To this end, we train our model in a teacher-student setup, learning to detect anomalies by distilling knowledge from multiple highly accurate object-centric teachers. Aside from the conventional distillation process based on the mean squared error between the low-resolution anomaly maps of the student and each teacher, we introduce an adversarial discriminator to distinguish between target (teacher) and reproduced (student) anomaly maps, as shown in Figure 1. To the best of our knowledge, we are the first to propose an adversarial knowledge distillation framework for anomaly detection in video.

We conduct experiments on three benchmark data sets: Avenue (Lu et al., 2013), ShanghaiTech (Luo et al., 2017) and UCSD Ped2 (Mahadevan et al., 2010). Our empirical evaluation shows that our method is 28 to 62 times faster compared with object-centric models (1480 FPS versus 24-52 FPS), while obtaining performance levels that are comparable to many recent methods (Astrid et al., 2021a,b; Barbălău et al., 2023).
Doshi and Yilmaz (2020), Ionescu et al. (2019a, b, 2021), Park et al. (2020), Ramachandra et al. (2020), Sun et al. (2020), Tang et al. (2020), Yu et al. (2022a) and less than 5% below the object-centric teachers (Georgescu et al., 2021, 2022b). As shown in Figure 2, our model achieves an excellent trade-off between speed and accuracy. In addition, we carry out a comprehensive ablation study to justify our architectural design choices.

In summary, our contribution is fourfold:

- We propose a novel teacher-student framework for anomaly detection in video, learning to detect anomalies by distilling from multiple highly accurate object-level teachers into a highly efficient student, at multiple scales.
- We propose adversarial knowledge distillation in the context of anomaly detection, introducing an adversarial discriminator for each teacher to distinguish between original (teacher) and reproduced (student) outputs.
- We carry out comparative and ablation experiments to assess the performance and speed gains of our design.
- To further increase the processing speed, we replace the fully connected layers inside the transformer blocks forming the student with pointwise convolutional layers.

2. Related work

Most frameworks formulate anomaly detection as a one-class learning problem, where only normal data is available at training time, while both normal and abnormal examples are present at test time (Pang et al., 2021, Ramachandra et al., 2022). The anomaly detection approaches are typically categorized into dictionary learning methods (Cheng et al., 2015), Cong et al., 2011, Dutta and Banerjee, 2015, Lu et al., 2013), probabilistic models (Feng et al., 2017, Hinami et al., 2017, Saleh et al., 2013, Wu et al., 2010), distance-based approaches (Ionescu et al., 2019a, b, Ramachandra and Jones, 2020, Ramachandra et al., 2020, Sabokrou et al., 2017, 2018, Saligrama and Chen, 2012, Smeureanu et al., 2017), reconstruction-based methods (Fei et al., 2022, Gong et al., 2019, Hasan et al., 2016, Liu et al., 2018a, Nguyen and Meunier, 2019, Park et al., 2020, Ristea et al., 2022) and change detection frameworks (Del Giorno et al., 2016, Ionescu et al., 2017, Liu et al., 2018b, Pang et al., 2020). Anomaly detection methods can also be divided into frame-level (Ionescu et al., 2019b, Liu et al., 2018b, Lu et al., 2013, Mahadevan et al., 2010, Mehran et al., 2009, Sabokrou et al., 2017) and object-level (Babaláš et al., 2023, Doshi and Yilmaz, 2020, Georgescu et al., 2021, 2022b, Ionescu et al., 2019a, Liu et al., 2021, Wang et al., 2022, Yu et al., 2020) frameworks, according to the level at which the anomaly detection algorithm is applied. These two categories are detailed below.

**Frame-level methods.** Frame-level methods detect anomalies by taking entire video frames as input. Yu et al. (2022a) proposed a frame-level framework that employs the adversarial learning of past and future events to detect anomalies. Liu et al. (2018a) proposed a simple yet effective algorithm, which learns to reconstruct the next frame of a short video sequence. Considering that frame-level methods commonly rely on producing poor reconstructions for anomalies, generalizing to out-of-distribution samples, e.g., anomalies, is not desired. To reduce the generalization capability, researchers employed various techniques, such as adding memory modules (Gong et al., 2019) or masked convolutional blocks (Madan et al., 2023, Ristea et al., 2022). Although integrating additional modules into the network leads to accuracy gains, the procedure often comes with a significant efficiency penalty.

Focusing primarily on efficiency, we propose a novel adversarial distillation framework that teaches our efficient frame-level student to replicate two heavy object-level teachers (Georgescu et al., 2021, 2022b). By distilling powerful teachers, we can afford to employ a lightweight frame-level student, gaining between 28 and 62 times the speed of the teachers with just a small performance drop (less than 5%).

**Object-level methods.** Lately, several powerful algorithms have been developed for anomaly detection by estimating anomaly scores at the object level (Babaláš et al., 2023, Ionescu et al., 2019a, Georgescu et al., 2022b, 2021, Liu et al., 2021, Yu et al., 2020, Wang et al., 2022) rather than at the frame level. Since the abnormal examples can be described as unusual objects in a regular scene, the prior information from an object detector enables anomaly detection models to focus only on objects, boosting the performance by a significant margin. One of the first object-level models (Ionescu et al., 2019a) is based on applying a k-means clustering in the joint latent space of motion and appearance auto-encoders. Later, Georgescu et al. (2022b) extended the framework by introducing pseudo-anomalies during training and attaching classifiers to discriminate between normal and pseudo-abnormal samples. Georgescu et al. (2021) employed self-supervised multi-task learning at the object-level to detect anomalies in video. The approach is further developed in a study (Babaláš et al., 2023) that adds new proxy tasks and a transformer-based architecture. Wang et al. (2022) studied alternative proxy tasks based on solving spatial and temporal jigsaw puzzles. A more complex approach is proposed by Liu et al. (2021), which conditions the image prediction on optical flow reconstruction using a conditional Variational Auto-Encoder (VAE).

Object-level methods rely on running an object detector, which limits the processing time to the speed of the detector. Thus, efficiency is significantly impacted, as the speed of the detector is often much lower than that of the anomaly network (Ionescu et al., 2019a, Georgescu et al., 2021). To address this issue, we propose a frame-level framework that is trained to reproduce the output of two strong object-level teachers (Georgescu et al., 2022b, 2021), eliminating the object detector from the equation during inference.

**Methods addressing efficiency.** While video anomaly detection has been extensively studied in recent years, the top scoring methods have certain speed limitations, preventing the deployment of state-of-the-art approaches in real-world scenarios, e.g., in video surveillance of entire cities. Even so, there is limited research on efficient anomaly detection frameworks (Doshi and Yilmaz, 2021, Fang et al., 2020, Li et al., 2021, Park et al., 2020).
and an acceptable trade-off between speed and performance has not been achieved yet. [Doshi and Yilmaz (2021)] proposed a video anomaly detection system that is capable of running on roadside cameras, while [Li et al. (2021)] proposed a neural architecture that learns normal behavior without supervision.

In our work, we design a lightweight hybrid convolutional- transformer student network that is trained to mimic the output of highly complex and accurate teachers, in an adversarial fashion. The limited size of our student leads to processing more than 1480 frames per second (FPS), while achieving competitive accuracy. Compared with competing methods, we achieve a significantly better speed versus performance trade-off.

**Transformers in anomaly detection.** Transformers have been widely adopted in computer vision, due to the outstanding results across a broad variety of tasks, ranging from object recognition [Dosovitskiy et al., 2021; Wu et al., 2021] and object detection [Carion et al., 2020; Zhu et al., 2020] to image generation [Ristea et al., 2023; Xu and Xu, 2022]. Distinct from approaches using only transformer-based attention [Carion et al., 2020; Chen et al., 2021; Dosovitskiy et al., 2021; Parmar et al., 2018; Ristea et al., 2023; Vaswani et al., 2017; Wu et al., 2021; Zhu et al., 2020], we propose a hybrid model, composed of both convolutional and self-attention blocks, to reduce the usually high processing time of transformers. More precisely, we improve the efficiency by downsampling the input and by replacing fully connected layers with pointwise convolutional layers.

To the best of our knowledge, there are only a handful of works applying transformers to video anomaly detection (Bårblåu et al., 2023; Jin et al., 2022; Madan et al., 2023; Chen et al., 2022). Unlike these related methods, which barely discuss the efficiency problem, our work is the first to significantly improve the accuracy versus efficiency trade-off of transformers for anomaly detection.

**Knowledge distillation in anomaly detection.** As a strategy for compressing one or multiple highly complex models, knowledge distillation [Ba and Caruana, 2014; Hinton et al., 2014] aims to efficiently train a lightweight neural network, called student, under the supervision of one or more deeper neural networks, called teachers, achieving competitive performance with fewer computational resources. Knowledge distillation was recently adopted for various vision tasks (Gou et al., 2021), e.g., object recognition [Chung et al., 2020; Park et al., 2019; Yu et al., 2019], face recognition [Feng et al., 2020; Georgescu et al., 2022a] and anomaly detection in industrial images [Bergmann et al., 2020; Cheng et al., 2021; Deng and Li, 2022; Salehi et al., 2021]. Comparatively lower attention has been devoted to applying knowledge distillation to video anomaly detection (Georgescu et al., 2021; Wang et al., 2021). Wang et al. (2021) employed a self-taught system based on a teacher-student model that does not require human-annotated data. Similarly, Georgescu et al. (2021) employed knowledge distillation and integrated it with other self-supervised tasks. The methods employing knowledge distillation for image or video anomaly detection leverage the representation discrepancy of anomalies between the teacher and the student models, which requires running both teacher and student models during inference.

Unlike the aforementioned methods, we perform knowledge distillation from teachers designed for video anomaly detection, specifically aiming to improve the computational time, running only the student model at test time. Moreover, to the best of our knowledge, we are the first to introduce adversarial knowledge distillation in the context of anomaly detection.

### 3. Method

**Overview.** We propose to train a fast and competitive frame-level model for video anomaly detection via multi-scale distillation. The training process is divided into two phases. In the first training phase, we attach a decoder to our network and train it as an auto-encoder to reconstruct normal training data, in a similar manner to previous studies (Lee et al., 2019; Liu et al., 2018a). Then, we diverge from the commonly-used auto-encoder model by discarding the decoder, attaching three anomaly detection heads to the model and training it via knowledge distillation. We employ two strategies to distill the knowledge from two object-level anomaly detection teacher models (Georgescu et al., 2021; 2022b) into our student, as illustrated in Figure 1. The first strategy is based on minimizing the mean squared error between the target anomaly maps provided by the teachers and the anomaly maps predicted by the student. The second strategy is based on adversarial training, relying on a neural discriminator that is optimized to distinguish between student and teacher anomaly maps. At the same time, our student is trained with an adversarial loss, trying to fool the discriminator. In this training setting, the student becomes a conditional generator.

An anomaly map is a matrix containing anomaly scores, which indicates the location of anomalies in a video frame. The teachers detect anomalies at the object level. To construct the teacher anomaly maps, we start from a map initialized with zeros, which is the same size as the corresponding video frame. For each detected object, we fill the corresponding bounding box in the anomaly map with the anomaly score predicted by a teacher. Following (Georgescu et al., 2021; 2022b), we keep the maximum anomaly score when two bounding boxes overlap. Finally, a Gaussian blur filter is applied to smooth the anomaly maps. These anomaly maps are downsampled at multiple scales (using bilinear interpolation) and used as targets for the student model. Our student architecture takes video frames as input and directly predicts the anomaly maps. This allows the student to bypass the costly object detection step. We next present the architecture and the training procedure of our student network.

**Student architecture.** As depicted in Figure 3 the architecture of our student network starts with a downsampling block which reduces the spatial dimensions of the input frames, thus speeding-up the subsequent processing. The second part of the architecture is formed of $m$ adapted convolution vision transformer (CvT) (Wu et al., 2021) blocks. The adaptation consists of replacing all dense layers with pointwise convolutional layers to further speed up the computation. The last part incorporates three multi-scale downsampling blocks generating anomaly maps at various resolutions.
Our initial downsampling block includes five convolutional layers, each followed by batch normalization and Rectified Linear Units (ReLU) (Nair and Hinton 2010). Let \( P \in \mathbb{R}^{h \times w \times c} \) denote the output tensor of the downsampling block, where \( h, w \) and \( c \) represent the height, width and number of channels. \( P \) is further processed by a convolutional transformer. The transformer encloses \( m \) consecutive transformer blocks and interprets \( P \) as a set of overlapping visual tokens. The transformer blocks follow the implementation of [Wu et al., 2021], where, prior to the self-attention mechanism, the queries \( Q \), the keys \( K \) and the values \( V \) are computed from \( P \) via convolutional projection. The next stage in the transformer block is the self-attention mechanism, which computes a new value vector for each token as a weighted sum of all value vectors in the set, where the weights are the attention scores. Formally, the operation is defined as follows:

\[
Z = \text{softmax} \left( \frac{Q \cdot K^\top}{\sqrt{d_q}} \right) \cdot V, \tag{1}
\]

where \( d_q \) is the size of a query token, \( K^\top \) is the transpose of \( K \), and \( Z \) is a matrix containing the new value vectors. We reshape \( Z \) into a tensor of \( h \times w \times d \) components. Our network comprises a multi-head attention module comprising \( h \) heads. We concatenate the tensors returned by all heads along the channel axis, obtaining a tensor that is further passed through a pointwise convolutional layer reducing the number of activation maps to the number of channels \( c \). The tensor \( P \) goes through a skip connection, being added to the resulting output, denoted as \( Z' \).

To significantly improve efficiency, we replace the multi-layer perceptron that typically follows in a conventional transformer block with two pointwise convolutional layers. The first pointwise convolutional layer comprises \( 4 \cdot c \) filters, and the second one only \( c \) filters.

The output of the convolutional transformer is fed into three downscaling heads returning anomaly maps at various resolutions. Each anomaly detection head is formed of two convolutional layers, each with kernels of \( 3 \times 3 \) and ReLU activations. The first conv layer is formed of 64 filters. The second conv layer comprises only one filter, and is followed by a pooling layer which returns the output anomaly map at the desired resolution. At inference time, we aggregate the anomaly maps returned by the three heads into a single frame-level anomaly score by computing the maximum value for each map, and averaging the maximum values.

**Student pre-training.** Our main objective is to transfer the anomaly detection capabilities from teacher networks to our student model. However, if we start the knowledge distillation with random weights for the student, the network must concurrently learn a high-level representation of the training scenes, and replicate the anomaly detection abilities of the teachers. In this scenario, the objective becomes significantly more difficult to complete. Therefore, we propose to independently address the scene understanding task by training the student as an auto-encoder to reconstruct input video frames, before introducing knowledge distillation. More precisely, we train the network to reconstruct the middle frame of a temporal sequence of frames. We fix the length of the temporal sequence to three frames, taken at a stride of \( t \). Given an input sequence \( \{x_i^{\text{ref}}, x_i', x_i^{\text{ref}}\} \) and the corresponding output of the auto-encoder \( \hat{x}_i' = D\left( E\left( \{x_i^{\text{ref}}, x_i', x_i^{\text{ref}}\} \right) \right) \), the optimization objective is defined as follows:

\[
\mathcal{L}_\text{AE} = \frac{1}{h \cdot w \cdot c} \sum_{j=1}^{h} \sum_{k=1}^{w} \sum_{l=1}^{c} \left( x_{ijkl}^{\text{ref}} - \hat{x}_{ijkl}' \right)^2, \tag{2}
\]

where \( E \) and \( D \) represent the encoder and decoder networks. Following [Ionescu et al., 2019a], we set \( t = 3 \) as the temporal step for the input sequence. The encoder \( E \) is formed of the initial downsampling block and the convolutional transformer.
(the multi-scale downsampling blocks are added later, during distillation). The decoder \( D \) is composed of transposed convolutions, being symmetric with the downsampling block of the encoder.

**Standard knowledge distillation.** As the first teacher, denoted as \( T_1 \), we choose the framework based on object-level motion and appearance auto-encoders of Georgescu et al. (2022b). We select the 3D object-centric network based on self-supervised multi-task learning (Georgescu et al., 2021) as the second teacher, denoted as \( T_2 \). The chosen teacher networks (Georgescu et al., 2021, 2022b) predict the anomaly score of each object detected by a pre-trained detector. Hence, an anomaly map containing the anomaly score associated with each bounding box can be obtained for each video frame. Consequently, our first distillation strategy is to train the student network to regress to the anomaly maps of the teachers, thus learning to imitate the teachers. To make the imitation task more accessible for the student, we lower the resolution of the target anomaly maps by passing them through a max pooling layer. Additionally, we employ multiple resolutions for the anomaly maps, allowing our student model to deal with objects seen at different scales. Let \( x_T \) and \( x_S \) be the input sequences of frames (centered on some frame) for the teacher and the student. Let \( r \) be the number of output resolutions, and \( y = T(x_T) \) and \( \hat{y} = S(x_S) \) the anomaly maps of some teacher \( T \) and the student \( S \), respectively. For some teacher \( T \), we optimize the following loss function:

\[
L_{\text{KD}}(T, S) = \sum_{k=1}^{r} \frac{1}{h_k \cdot w_k} \sum_{i=1}^{h_k} \sum_{j=1}^{w_k} (y_{ij} - \hat{y}_{ij})^2, \tag{3}
\]

where \( h_k \) and \( w_k \) are the height and the width of the anomaly maps \( y \) and \( \hat{y} \) at the \( k \)-th resolution. In practice, we set the number of output resolutions to \( r = 3 \).

Eq. (3) expresses the loss with respect to only one teacher network. However, to increase the generalization power of the student, we distill knowledge from two different anomaly detection models, denoted as \( T_1 \) and \( T_2 \). To this end, the objective becomes:

\[
L_{\text{KD}} = L_{\text{KD}}(T_1, S) + L_{\text{KD}}(T_2, S). \tag{4}
\]

**Adversarial distillation.** Our student is applied directly at the frame level, which means that it needs to learn to jointly detect objects and predict anomalies in a single pass, having a significantly more difficult task than its teachers. Thus, to better mimic the behavior of the teachers, we leverage the use of adversarial training, a procedure typically employed to train generative adversarial networks (GANs) (Goodfellow et al. 2014). For each teacher involved in the distillation, we introduce a discriminator trained to label the anomaly maps produced by the student as fake examples (class 0) and the ones produced by the teacher as genuine examples (class 1). As mentioned previously, the student outputs multi-resolution anomaly maps. Each discriminator receives all anomaly map variants and concatenates them (at the appropriate levels) into a single neural network. Therefore, each discriminator makes a single prediction for all anomaly map resolutions.

Let \( D_T \) be the discriminator corresponding to a teacher \( T \). We train \( D_T \) and \( S \) through a zero-sum game, alternating between minimizing (for \( S \)) and maximizing (for \( D_T \)) the following loss:

\[
L_{\text{AKD}}(D_T, S) = E_{x_T \sim p_T}[\log D_T(T(x_T))] + E_{x_S \sim p_S}[\log (1 - D_T(S(x_S)))]. \tag{5}
\]

where \( x_T \) and \( x_S \) are input sequences of frames for the teacher and the student models, and \( p_T \) and \( p_S \) are data densities producing samples for the teacher and the student, respectively. We hereby underline that the teacher model \( T \) is frozen. In practice, the data densities \( p_T \) and \( p_S \) are identical and \( x_T = x_S \).

To encapsulate both teachers in the objective formulation, we need to apply Eq. (5) for each discriminator and sum the corresponding losses, as follows:

\[
L_{\text{AKD}} = L_{\text{AKD}}(D_{T_1}, S) + L_{\text{AKD}}(D_{T_2}, S). \tag{6}
\]

where \( D_{T_1} \) and \( D_{T_2} \) are the discriminators of teachers \( T_1 \) and \( T_2 \), respectively.

**Discriminator architecture.** We employ a convolutional neural network (CNN) as adversarial discriminator, structured into three distinct blocks, as depicted in Figure 3. The initial block processes the anomaly map at the highest resolution, and its output is merged with the anomaly map at the middle resolution to form the input for the subsequent block. Similarly, the output from the second block is combined with the anomaly map at the lowest resolution and fed into the final block. The convolutional blocks share a uniform architecture, namely a convolutional layer succeeded by a max-pooling layer. The first two blocks are based on convolutional layers with \( 3 \times 3 \) filters applied at a stride of 1, using a zero-padding of 1 pixel in all directions. All units are activated by ReLU. The first conv layer contains 16 filters and the second is equipped with 32 filters. The third block is based on a dense layer of 100 neurons activated by ReLU. A final Softmax layer determines whether the detected anomalies originate from a teacher or a student model.

**Joint distillation.** The final objective of our framework combines the losses defined in Eq. (4) and Eq. (6) into a single loss function as follows:

\[
L_{\text{total}} = L_{\text{KD}} + \alpha \cdot L_{\text{AKD}}, \tag{7}
\]

where the adversarial loss receives a fixed weight \( \alpha = 0.1 \).

4. **Experiments**

4.1. **Data sets**

We conduct experiments on three of the most popular data sets for video anomaly detection: Avenue (Lu et al., 2013), ShanghaiTech (Luo et al., 2017) and UCSD Ped2 (Mahadevan et al., 2010).

**Avenue.** The Avenue data set (Lu et al. 2013) includes 16 training videos (15,328 frames) of normal events, and 21 test videos (15,324 frames) with both normal and abnormal situations. The total number of video frames is 30,652.

**ShanghaiTech.** The ShanghaiTech data set (Luo et al. 2017) is a much larger data set, containing 330 training videos of normal
events, and 107 test videos with normal and abnormal scenes. The total number of frames is about 320K, with over 270K being used for training.

**UCSD Ped2.** UCSD Ped2 (Mahadevan et al., 2010) contains 16 training videos and 12 test videos. As the other two data sets, the training videos depict only normal scenarios.

### 4.2. Experimental setup

**Teachers.** Our frame-level network is trained to replicate the output of two highly effective object-level teachers (Georgescu et al., 2022b) designed for video anomaly detection. Each teacher detects objects with a pre-trained YOLOv3 (Redmon and Farhadi, 2018). The objects are further processed by an object-centric anomaly model. We use the official code to train the teachers.

Georgescu et al. (2022b) presents a comprehensive framework comprising an object detector, an optical flow estimator, and three auto-encoders, each dedicated to a distinct reconstruction task. Two of these auto-encoders are tasked with reconstructing forward and backward motion, while the third focuses on reconstructing the appearance. The architecture of each encoder is based on three convolutional layers. The structure of each decoder is organized into three blocks, with each block consisting of upsampling and convolution operations, mirroring the encoding architecture. It is important to note that the overall time efficiency of this method, designated as our first teacher, is significantly influenced by the speeds of the object detector and the optical flow estimator, respectively.

Georgescu et al. (2021) proposes an object-centric method, which we choose to serve as the second teacher for our student net. The detected objects are fed into a 3D CNN that serves as a feature extractor, followed by four prediction modules. Three of these modules utilize a simple architecture, comprising one convolutional layer followed by a fully-connected layer. The fourth module, designed to estimate the masked central bounding box in an object-centric sequence, mirrors the structure of the 3D CNN, but employs 2D convolutional layers to accommodate its specific predictive task. The authors explore several configurations of the 3D CNN, with the most comprehensive and effective version featuring nine convolutional layers. The width of these layers varies, with the number of filters ranging from 32 to 64.

We underline that our frame-level student uses a transformer-based architecture, while the teachers are based on complex frameworks comprising multiple networks. For instance, the teachers employ a YOLO-based model to detect objects, and convolutional architectures to label objects with anomaly scores. In contrast, our student uses a single end-to-end network, thus having a simplified and efficient structure.

**Baselines.** We compare our method with state-of-the-art frame-level (Astrid et al., 2021a; Gong et al., 2019; Ionescu et al., 2019a; Nguyen and Meunier, 2019; Park et al., 2020; Ramachandra and Jones, 2020; Ravanbakhsh et al., 2018; Sun et al., 2018; Tang et al., 2020; Wu et al., 2019; Wu et al., 2020; Wu et al., 2022; Yu et al., 2022; Zhang et al., 2020) and object-level (Barbalau et al., 2023; Doshi and Yilmaz, 2020; Georgescu et al., 2021; Madan et al., 2023; Ristea et al., 2022; Wang et al., 2022; Yu et al., 2020) frameworks.

**Hyperparameters.** We opt for a lightweight model having under 5M parameters (as reference, ResNet-18 has 11M parameters), setting the number of transformer blocks to $m = 5$, the number of self-attention heads to $s = 5$, and the number of output heads to $3$. For Avenue and ShanghaiTech, the output resolutions are $1 \times 1$, $4 \times 4$ and $16 \times 16$. Since the objects in UCSD Ped2 have a much smaller scale, the output resolutions are set to $1 \times 1$, $30 \times 30$ and $60 \times 60$. Regardless of the data set, the student network is trained for 35 epochs on mini-batches of 64 video sequences of 3 frames each. We optimize the model with Adam (Kingma and Ba, 2015), using a learning rate of $10^{-4}$ and a weight decay set to $10^{-5}$ (all other hyperparameters are set to default values).

**Evaluation.** Following recent works (Ascientoae et al., 2022; Georgescu et al., 2022b; Ristea et al., 2022), we evaluate detection performance in terms of the micro-averaged and macro-averaged frame-level AUC, and localization performance in terms of the Region-Based Detection Criterion (RBDC) and Track-Based Detection Criterion (TBDC). For both AUC measures, we compute the area under the ROC curve (AUC) with respect to the ground-truth frame-level annotations. A frame is labeled as abnormal if the predicted anomaly score is larger than the current threshold. For the macro AUC, all the test frames are first concatenated into a single video before computing the AUC. For the micro AUC, we compute the AUC of each test video and report the mean of the resulting AUC scores. As recommended by Ramachandra and Jones (2020), we set the region overlap threshold to 0.1 for RBDC, and the track overlap threshold to 0.1 for TBDC.

### 4.3. Results

#### 4.3.1. Quantitative analysis

In Table 1, we present the micro and macro AUC scores of our method in comparison with the state-of-the-art frame-level and object-level methods on the Avenue, ShanghaiTech and UCSD Ped2 benchmarks. Even if our method is below the top scoring object-centric methods, it yields comparable performance with some methods from this category, e.g. (Ionescu et al., 2019a; Yu et al., 2020). Moreover, with a micro AUC of 88.3%, our student outperforms most of the frame-based models. Among all frame-level frameworks reporting results on ShanghaiTech, we observe that our model ranks first in terms of the macro AUC, and third in terms of the micro AUC. With respect to the teachers (Georgescu et al., 2021, 2022b), our micro and macro AUC drops are lower than 5%. In terms of the macro AUC on UCSD Ped2, our model is just below the state-of-the-art object-centric approach (Georgescu et al., 2022b), at a difference of only 2.7%. In terms of the micro AUC, the student registers a performance drop of at most 2.8% with respect to both teachers (Georgescu et al., 2021, 2022b).

---

https://github.com/lillygeorgescu/AED
https://github.com/lillygeorgescu/AED-SSMTL
In Table 2, we compare the RBDC and TBDC scores of our method with the RBDC and TBDC scores of other studies which reported these localization metrics. In general, frame-level methods are less able to localize anomalies than object-centric methods. This explains why the number of frame-level methods reporting the RBDC and TBDC scores is much lower than the number of object-centric methods. Notably, our method reaches the highest RBDC scores among all frame-level methods, on both Avenue and ShanghaiTech. While our TBDC score is not competitive on the Avenue data set, we attain the highest TBDC score on ShanghaiTech, when our method is compared with methods from the same category. Furthermore, we observe that our method also surpasses the object-centric method of Ionescu et al. (2019a). We conclude that the localization performance of our approach is competitive.

4.3.2. Qualitative analysis

Frame-level scores. We provide frame-level output visualizations on multiple test videos. In Figure 4, we illustrate the anomaly detection performance on test video 14 from Avenue. Our student is able to surpass the lower results of teacher $T_1$, being very close to teacher $T_2$. This shows the importance of learning from more than one teacher. In Figure 5, we present the anomaly detection performance on test video 1 from Avenue. On this video, our student is remarkably able to surpass both teachers by more than 3% in terms of the AUC.
| Type                | Method                                      | Avenue RBDC | Avenue TBDC | Shanghai RBDC | Shanghai TBDC | FPS |
|---------------------|---------------------------------------------|-------------|-------------|----------------|----------------|-----|
|                     | Barbălău et al. (2023)                      | 47.83       | 85.26       | -              | -              | 20  |
|                     | Georgescu et al. (2021)                     | 57.00       | 58.30       | 42.80          | 83.90          | 51  |
|                     | Georgescu et al. (2022b)                    | 65.05       | 66.85       | 41.34          | 78.79          | 24  |
|                     | Ionescu et al. (2019a)                      | 15.77       | 27.01       | 20.65          | 44.54          | -   |
|                     | Liu et al. (2021)                           | 41.05       | 86.18       | 44.41          | 83.86          | 12  |
|                     | Madan et al. (2023) + Barbălău et al. (2023) | 49.01       | 85.94       | 47.73          | 85.68          | 20  |
|                     | Madan et al. (2023) + Georgescu et al. (2022b) | 66.04       | 65.12       | 40.52          | 81.93          | 31  |
|                     | Madan et al. (2023) + Liu et al. (2021)     | 46.49       | 86.43       | 45.86          | 84.69          | 10  |
|                     | Ristea et al. (2022) + Georgescu et al. (2022b) | 65.99       | 64.91       | 40.55          | 83.46          | 31  |
|                     | Ristea et al. (2022) + Liu et al. (2021)    | 62.27       | 89.28       | 45.45          | 84.50          | 10  |
|                     | Liu et al. (2018a)                          | 19.59       | 56.01       | 17.03          | 54.23          | 28  |
|                     | Madan et al. (2023) + Liu et al. (2018a)    | 23.79       | 66.03       | 19.13          | 61.65          | 26  |
|                     | Ramachandra and Jones (2020)                | 35.80       | 80.90       | -              | -              | -   |
|                     | Ramachandra and Jones (2020)                | 41.20       | 78.60       | -              | -              | -   |
|                     | Ristea et al. (2022) + Liu et al. (2018a)   | 20.13       | 62.30       | 18.51          | 60.22          | 26  |
|                     | Ours                                         | 41.92       | 44.49       | 19.34          | 63.64          | 1480|

Table 2. RBDC and TBDC scores (in %) of our method versus several state-of-the-art frame-level and object-level methods on Avenue and ShanghaiTech. For each family of methods, the best score is shown in red, the second-best in green, and third-best in blue. The reported FPS values (including those of the baselines) are measured on a machine with an Nvidia GeForce GTX 3090 GPU with 24 GB of VRAM.

This highlights that, even if the complexity of the student network is significantly lower compared with the complexity of the teachers, the student is able to learn aggregated information, which can eventually lead to surpassing the teachers in certain cases. An even higher gap in favor of our student is illustrated in Figure 6. Indeed, the frame-level anomaly predictions on test video 01_0136 from ShanghaiTech indicate that our student surpasses the best teacher, $T_1$ (Georgescu et al. 2022b), by more than 12%, attaining an overall AUC of 97.46%. In Figure 7, we showcase the anomaly detection performance on test video 01_0055 from ShanghaiTech. Our student, being trained to mimic both teachers, is able to surpass the lower results of teacher $T_1$, getting close to the performance of teacher $T_2$. For the UCSD Ped2 data set, we showcase the frame-level scores on test video 3 in Figure 8. In this example, the student model attains an AUC score of 98.63%, being very close to the teachers, which have perfect scores. In Figure 9, we present the anomaly detection performance on test video 6 from UCSD Ped2. Here, the performance of our student is between that of the teachers $T_1$ and $T_2$.

Anomaly maps. In Figure 10, we illustrate a set of predicted versus ground-truth anomaly maps for five test frames from Avenue. The anomaly maps on the third row are predicted by our student model, while those on the fourth and fifth rows are pre-
dicted by the teachers $T_1$ and $T_2$, respectively. We underline that the anomaly maps generated by the student are smoother, because it does not rely on a pre-trained detector to obtain object bounding boxes. This feature seems to be extremely helpful for the example depicted on the second column, where the teachers fail to detect the papers thrown in the air, since paper is not among the classes known by YOLOv3 \cite{RedmonFarhadi2018}. In contrast, our student assigns high anomaly scores to the region where there are papers in the air. Another interesting behavior for our student is its ability to assign a high score to the child running in the example shown on the fourth column, without triggering high anomaly scores for the person standing on the grass, as opposed to the teachers. The example shown on the last column depicts a normal event, where people are just walking. Here, we observe that teacher $T_2$ outputs high anomaly scores without a good reason. In the end, we believe that one of the most remarkable abilities of our student is to detect anomalies where object-centric approaches based on detectors with a limited number of known classes fail.
4.4. Speed versus accuracy trade-off

In Figure 2, we compare our model with several recent frame-level and object-centric methods in terms of the trade-off between accuracy and speed, on the Avenue benchmark. The running times of all methods (Georgescu et al., 2021, 2022; Gong et al., 2019; Ionescu et al., 2017; Liu et al., 2018a, 2021; Park et al., 2020, 2022; Ristea et al., 2022; Wang et al., 2022), including our own, are measured on a machine with an Nvidia GeForce GTX 3090 GPU with 24 GB of VRAM. The FPS rates of the object-level methods range between 20 and 52. Some frame-level methods are comparatively faster, the most efficient one, proposed by Park et al. (2022), running at 195 FPS. Our method runs at 1480 FPS, being more than 7 times faster than the most efficient competitor (Park et al., 2022). Remarkably, we also outperform the fastest competitor (Park et al., 2022) in terms of the micro AUC (by 3%). In Figure 11, we present the same trade-off between accuracy and time on the ShanghaiTech benchmark, and we see that our method outperforms again the fastest competitor, exhibiting a superior micro AUC advantage of 5.8%.

Comparison with FastAno. The trade-offs reported on both Avenue and ShanghaiTech clearly indicate FastAno (Park et al., 2022) as the fastest competing model. On both data sets, our model obtains a better trade-off between speed and accuracy, being more than 7 times faster while reaching scores that are more than 5% better. The state-of-the-art trade-off of our model stems from our option to use a very lightweight architecture (having under 5 million parameters) that is applied on whole frames, using a downsampling block to quickly down-scale the input resolution. FastAno (Park et al., 2022) and most frame-level works divide the frames into patches or cuboids, which need to be processed as individual samples in mini-batches. Our results show that processing a single mini-batch of frames (with an aggressive downsampling block) instead of many mini-batches of patches is significantly more efficient and effective. This is made possible by our training procedure based on distilling knowledge from highly accurate object-centric teachers.

Comparison with teachers. We showed that our student model runs more efficiently on GPU compared with other models. To confirm that the speed gains are not specific to GPU environments, we tested our model and its teachers, $T_1$ and $T_2$, in a different environment (Intel i9 CPU 3.7 GHz, no GPU). On CPU, our model runs at 88 FPS, while the teachers $T_1$ and $T_2$ run at 2 and 4 FPS, respectively. These running times confirm the speed superiority of our model, which is still 22 to 44 times

| Usage Parameter          | Teacher $T_2$ | Student |
|--------------------------|---------------|---------|
| Memory usage (GB)        | 17            | 6.5     |
| Trainable parameters (M) | 65            | 5       |
| GFLOPs                   | 107.9         | 6.2     |
faster than the object-centric teachers. Remarkably, our model can process about 3 video streams at 30 FPS on CPU, while the teachers must rely on the GPU environment to deliver real-time processing for 1 or 2 streams. Thus, our student can process more streams on CPU than its teachers on GPU.

Aside from comparing our model with the teachers in terms of the FPS, we also look at other characteristics, such as the memory usage, the number of parameters and the number of floating point operations (GFLOPs). We choose to compare our student with teacher $T_2$ (Georgescu et al., 2021), since this teacher proved to be faster than teacher $T_1$ (Georgescu et al., 2022b) in both GPU and CPU environments. The comparison is shown in Table 3. Considering the additional usage parameters, we find that our student preserves its efficiency, consuming about one third of the RAM and using 17 times less GFLOPs. This is mainly due to our lighter architecture, which has 13 times less trainable weights.

4.5. Ablation studies

We start developing our model from a truncated architecture that takes a single frame as input and produces anomaly maps of $4 \times 4$ pixels. The truncated architecture is trained using only standard knowledge distillation from a single teacher, namely $T_1$ (Georgescu et al., 2022b).

We report both micro and macro AUC scores for our ablation studies. When making our design choices, we refer to the micro AUC as the more representative measure, due to its wider popularity (in contrast to the macro AUC) across video anomaly detection papers (Ramachandra et al., 2022).

Transformer architecture. To determine the optimal depth for our convolutional transformer, we first fix the number of heads to $s = 5$ and experiment with architectures comprising $m$ blocks, where $m \in \{3, 4, 5, 6, 7\}$. The results reported in Table 4 indicate that using $m = 5$ transformer blocks leads to the highest micro AUC score, while keeping an astonishing FPS rate of 1480. Next, we fix the number of blocks to $m = 5$ and variate the number of heads, considering $s \in \{3, 4, 5, 6, 7\}$. Unlike the number of blocks, we observe that the number of heads does not have a high impact on the running time. We obtain the highest micro AUC with $s = 5$ heads. We underline that all configurations are much faster than competing anomaly detection methods. We hereby focus on choosing the configuration providing the highest micro AUC for the subsequent experiments, namely the one with $m = 5$ blocks and $s = 5$ heads.

Output heads. The task of replicating the full-resolution anomaly maps produced by the state-of-the-art teacher models (Georgescu et al., 2021, 2022b) is quite difficult, and the anomaly detection performance is directly impacted by the ability of the student to replicate the teachers. To simplify the task, we reduce the resolution of the anomaly maps. To determine the optimal resolution, we test different output head configurations, reporting the results in Table 5. According to the presented results, our model seems to achieve better micro AUC scores when producing anomaly maps of $1 \times 1$ and $4 \times 4$ pixels. The micro AUC tends to drop as we increase the resolution of the anomaly maps. Aside from testing individual output heads, we also consider combining the low-resolution heads into a sin-

---

**Table 4.** Micro and macro AUC scores (in %) and FPS rates on the Avenue data set [Lu et al., 2013], while varying the number of transformer blocks $m$ and the number of heads $s$ inside our architecture.

| $m \times s$ | AUC       | FPS       |
|------------|-----------|-----------|
| 1 $\times$ 1 | Micro: 68.2 | Macro: 74.3 |
| 1 $\times$ 4 | Micro: 75.4 | Macro: 75.5 |
| 1 $\times$ 16 | Micro: 65.7 | Macro: 74.0 |
| 1 $\times$ 64 | Micro: 62.3 | Macro: 65.8 |
| 1 Full       | Micro: 59.2 | Macro: 73.9 |
| 2 $\times$ 1, 4 $\times$ 4 | Micro: 71.7 | Macro: 74.7 |
| 3 $\times$ 1, 4 $\times$ 4, 16 $\times$ 16 | Micro: 76.4 | Macro: 76.0 |

**Table 5.** Micro and macro AUC scores (in %) on the Avenue data set [Lu et al., 2013], while varying the resolution and the number of output heads. The top scores are highlighted in bold.
Table 6. Micro and macro AUC scores (in %) on the Avenue data set (Lu et al., 2013), while varying the number of input frames. The top scores are highlighted in bold.

| #Input Frames | AUC          |
|---------------|--------------|
|               | Micro | Macro |
| 1             | 76.4  | 76.0  |
| 3             | 79.5  | 80.7  |
| 5             | 78.4  | 80.3  |

Table 7. Micro and macro AUC scores (in %) on Avenue (Lu et al., 2013), while using different training procedures based on input reconstruction pre-training ($L_{AE}$), knowledge distillation ($L_{KD}$), and adversarial knowledge distillation ($L_{AKD}$). Results with one versus two teachers are also included.

| Teachers | $L_{AE}$ | $L_{KD}$ | $L_{AKD}$ | AUC          |
|----------|----------|----------|-----------|--------------|
|          |          |          |           | Micro | Macro |
| $T_1$    | ✓        | ✓        | ✓         | 72.6  | 73.2  |
| $T_1$    | ✓        | ✓        | ✓         | 79.5  | 80.7  |
| $T_1$    | ✓        | ✓        | ✓         | 63.7  | 69.8  |
| $T_1$    | ✓        | ✓        | ✓         | 85.0  | 86.5  |
| $T_1$    | ✓        | ✓        | ✓         | 69.1  | 80.6  |
| $T_1$    | ✓        | ✓        | ✓         | 82.4  | 82.9  |
| $T_1$    | ✓        | ✓        | ✓         | 86.2  | 86.7  |
| $T_1 + T_2$ | ✓       | ✓        | ✓         | 88.3  | 87.9  |

Table 8. Micro and macro AUC scores (in %) on Avenue (Lu et al., 2013), ShanghaiTech (Luo et al., 2017) and UCSD Ped2 (Mahadevan et al., 2010), for student models trained with two teachers, before and after adding adversarial distillation. Top scores are highlighted in bold.

| Teachers | $L_{AKD}$ | Avenue | Shanghai | Ped2 |
|----------|-----------|--------|----------|------|
|          | Micro | Macro | Micro | Macro | Micro | Macro |
| $T_1 + T_2$ | ✓    | 88.2  | 87.7  | 77.8  | 84.6  | 93.8  | 96.5  |
| $T_1 + T_2$ | ✓    | 88.3  | 87.9  | 78.0  | 86.0  | 95.9  | 97.1  |

Fig. 12. Micro and macro AUC scores (in %) on the Avenue data set (Lu et al., 2013) for distinct values of the adversarial knowledge distillation weight factor $\alpha$. 

Adversarial loss weight. Throughout all the experiments involving adversarial knowledge distillation, the hyperparameter controlling the importance of the adversarial distillation loss in Eq. (7) is set to $\alpha = 0.1$. However, to determine if the proposed loss is robust to variations of $\alpha$, in Figure 12 we present additional results with values of $\alpha$ between 0.1 and 0.7, taken at a step of 0.1. The results for $\alpha = 0$ correspond to the model trained without adversarial knowledge distillation. As indicated by the results illustrated in Figure 12 there are multiple values for $\alpha$, between 0.1 and 0.5, leading to superior results compared with the baseline based only on standard knowledge distillation (which uses $\alpha = 0$). We thus conclude that the loss proposed in Eq. (7) is robust to variations of $\alpha$, allowing us to set this.
Multiple teachers and adversarial training. The ablation results presented so far distill knowledge from a single teacher, namely $T_1$ (Georgescu et al., 2022b). Since different models can often produce distinct anomaly maps, it can be reasonably argued that combining the respective models can lead to performance improvements, an argument which forms the basis of ensemble learning (Rokach, 2010). We conjecture that the same principle applies to knowledge distillation, i.e., distilling knowledge from multiple teachers is likely to bring performance gains. To this end, we propose to distill knowledge from two powerful teachers, denoted as $T_1$ (Georgescu et al., 2022b) and $T_2$ (Georgescu et al., 2021). In Table 9 we present results for alternating between individual and combined teachers, applying standard knowledge distillation at first. Our empirical results confirm that introducing two teachers leads to superior micro and macro AUC scores.

We next turn our attention to introducing adversarial knowledge distillation with multiple teachers, which is not trivial. We study three possible ways of integrating multiple teachers with adversarial distillation:

- **One adversarial discriminator, one output branch.**
  In this configuration, the student has a single multi-resolution output branch for both teachers, and the adversarial discriminator distinguishes between three categories of anomaly maps produced by the student, the teacher $T_1$, and the teacher $T_2$, respectively. Within this framework, adding more teachers requires adding an equal number of adversarial discriminators.

- **Two adversarial discriminators, two output branches.**
  In this framework, the student has a separate multi-resolution output branch and an associated adversarial discriminator for each teacher $T_i$. Since there are multiple output branches for this framework, we need a method to aggregate the anomaly maps at a certain resolution. We consider taking the element-wise maximum or the average of the corresponding anomaly maps, reporting the results with the better aggregation function only, namely the element-wise maximum.

In the second part of Table 9 we present the results with the aforementioned frameworks for integrating multiple teachers with adversarial distillation. Among the three alternative frameworks, the best results are obtained by the approach comprising two adversarial discriminators and one output branch. However, it appears that the other two frameworks are not able to surpass the results obtained by distilling both teachers with standard knowledge distillation. In summary, there seems to be only one way to go forward: using two adversarial discriminators and one output branch. Still, the performance gains of the only viable approach on Avenue seem marginal. In Table 8 we have already extended this analysis to ShanghaiTech and UCSD Ped2, observing that combining adversarial distillation and multiple teachers brings higher gains on the other benchmarks. All in all, these results confirm that it requires meticulous design and significant effort in formulating a unified framework that is both efficient and effective.

**Teacher variations.** Upon establishing how to integrate adversarial distillation with multiple teachers, we test additional hypotheses regarding the number and the diversity of teachers. We report the corresponding results in Table 10. First, we carry out an experiment with two runs of teacher $T_1$ on Avenue, aiming to verify if the diversity among teachers is important. We obtain inferior performance when distilling two runs of teacher $T_1$ in-

| Teacher | #Adv. Disc. | #Output Branches | Micro AUC | Macro AUC |
|---------|--------------|------------------|----------|----------|
| $T_1$   | 0            | 1                | 86.2     | 86.7     |
| $T_2$   | 0            | 1                | 82.1     | 85.5     |
| $T_1 + T_2$ | 0 | 1                | 88.2     | 87.7     |

| Teacher | #Adv. Disc. | #Output Branches | Micro AUC | Macro AUC |
|---------|--------------|------------------|----------|----------|
| $T_1$   | 1            | 1                | 85.9     | 86.2     |
| $T_1 + T_2$ | 2 | 1                | 88.3     | 87.9     |
| $T_1 + T_2$ | 2 | 2                | 86.4     | 86.0     |

Table 9. Micro and macro AUC scores (in %) on the Avenue data set (Lu et al., 2013), while alternating between individual and combined teachers. Results for different attempts of performing adversarial distillation with multiple teachers are also reported. The top scores are highlighted in bold.

| Architecture | AUC Micro | AUC Macro | FPS |
|--------------|-----------|-----------|-----|
| CNN + CvT (standard, with fc) | 71.8 | 77.1 | 1267 |
| CNN + CvT (pointwise conv, no fc) | **85.0** | **86.5** | **1480** |

Table 10. Micro and macro AUC scores (in %) and FPS rates on the Avenue data set (Lu et al., 2013), while alternating between CvT blocks with fully connected (fc) layers (standard) or pointwise convolutions (proposed). The top scores are highlighted in bold.
The depthwise convolution has 256 filters with a spatial dimension. This operation is a depthwise separable convolution with each block containing one convolutional projection and a self-attention mechanism [Wu et al., 2021].

Next, we conduct another experiment on Avenue by adding a new teacher T2 (Liu et al., 2021) to our lineup of teachers. The teacher developed by Liu et al. (2021) stimulates our student to reach even better results. However, the performance gains are not as high as going from one teacher to two teachers. Therefore, it is likely to observe a saturation of the improvements after a certain number of teachers.

**Dense layers versus pointwise convolutions.** In Table 11, we demonstrate the effectiveness of replacing the last two fully connected layers from the CvT block (Wu et al., 2021) with pointwise convolutions. For a fair comparison, both model variants are trained with middle frame reconstruction in the first training stage and standard knowledge distillation in the second stage. Aside from the expected speed improvement, we observe that the proposed change also generates significantly better micro and macro AUC scores. In conclusion, the empirical evaluation clearly indicates that replacing fully connected layers with pointwise convolutions is beneficial in terms of both speed and accuracy.

While replacing the multi-layer perceptron with pointwise convolutions might seem as a natural choice, to our knowledge, this has not been proposed before within anomaly detection transformers. We consider this contribution very important due to its accuracy and speed gains (+13.2% micro AUC and +213 FPS).

5. Reproducibility

To ensure reproducibility, our source code is publicly released at https://github.com/risoea/fast-aed. With the same goal in mind, we next present the student architecture in more detail.

Our student comprises a convolutional downsampling block, a sequence of transformer blocks, and a set of output blocks. The downsampling block has five convolutional layers. The first convolutional layer is formed of 16 filters with a spatial size of 7 × 7 and no padding. The subsequent convolutional layers of the downsampling block have 32, 64, 128 and 256 filters, respectively, each with a kernel size of 3 × 3. Each layer uses a stride of 2 and a padding of 1.

The transformer encloses m consecutive transformer blocks and interprets P as a set of overlapping visual tokens. The transformer blocks follow the implementation of Wu et al. (2021), where, prior to the self-attention mechanism, the queries Q, the keys K and the values V are computed from P via convolutional projection. This operation is a depthwise separable convolution, being implemented with different weights for Q, K and V. The depthwise convolution has 256 filters with a spatial dimension of 3 × 3. It is applied at a stride of 2 for K and V, and a stride of 1 for Q. This layer is followed in all three cases by a batch normalization layer and a pointwise convolution with 64 filters. The final matrices \( Q \in \mathbb{R}^{n_q \times d}, K \in \mathbb{R}^{n_k \times d} \) and \( V \in \mathbb{R}^{n_v \times d} \) are obtained after flattening the activation maps, while preserving the channel dimension. We note that \( n_q = h \cdot w, n_k = n_v = \frac{h \cdot w}{m} \) and \( d = 64 \).

The next stage in the transformer block is the self-attention mechanism defined as follows:

\[
Z = \text{softmax} \left( \frac{Q \cdot K^\top}{\sqrt{d}} \right) \cdot V, \quad (8)
\]

where \( K^\top \) is the transpose of \( K \), and \( Z \) is a matrix of size \( n_q \times d \) containing the new value vectors. We reshape \( Z \) into a tensor of \( h \times w \times d \) components.

Our network comprises a multi-head attention module, each head containing one convolutional projection and a self-attention mechanism. The tensors resulting from all heads are concatenated along the channel axis, obtaining a tensor \( Z^* \) of \( h \times w \times (d \cdot s) \) components, where \( s \) is the number of heads. The tensor \( Z^* \) is further passed through a pointwise convolutional layer that reduces the number of activation maps to the number of channels \( c \). The tensor \( P \) goes through a skip connection, being summed up with \( Z^* \). The resulting output is denoted by \( Z^\prime \).

The final output of the multi-head attention module is fed into a batch normalization layer. To significantly improve efficiency, we replace the multi-layer perceptron that typically follows in a conventional transformer block with two pointwise convolutional layers. The first pointwise convolutional layer comprises \( 4 \cdot c \) filters, and the second one only \( c \) filters. Another skip connection is added between the input \( Z^\prime \) and the output of this module.

6. Limitations

Although we achieved a very good trade-off between efficiency and performance in this work, we admit that there is an observable gap between the performance of our model and that of the best-performing object-centric method. Moreover, since our method is based on knowledge distillation, we also need to apply the teachers and obtain target anomaly maps, whenever we want to train our method on a new dataset. In future work, our objective is to enhance performance, which can be pursued by introducing more proxy tasks for the student, in compliance with those used by the teachers. Moreover, investigating the potential of pre-training approaches via self-supervised tasks, in conjunction with various architectural configurations, represents an additional compelling direction of future research.

7. Conclusion

In this paper, we introduced a novel teacher-student framework for anomaly detection in video, learning to detect anomalies by distilling knowledge from multiple highly accurate object-level teachers into a highly efficient student. Moreover, we proposed adversarial knowledge distillation in the context of anomaly detection, introducing an adversarial discriminator for each teacher. Using our novel framework, we successfully
trained an efficient convolutional transformer, obtaining a state-of-the-art speed versus performance trade-off. More precisely, we reached an unprecedented speed of 1480 FPS, with a minimal performance decrease.

In future work, we aim to further raise the accuracy of our student by introducing more teachers and tasks.

Acknowledgements

This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS - UEFISCEDI, project number PN-III-P2-2.1-PED-2021-0195, within PNCDI III.

References

Acisintaoae, A., Florescu, A., Georgescu, M., Mare, T., Sumedrea, P., Ionescu, R.T., Khan, F.S., Shah, M., 2022. Unnormal: New benchmark for supervised open-set video anomaly detection, in: Proceedings of CVPR, pp. 20143–20153. Astrid, M., Zaheer, M.Z., Lee, J.Y., Lee, S.I., 2021a. Learning not to reconstruct anomalies, in: Proceedings of BMVC. Astrid, M., Zaheer, M.Z., Lee, S.I., 2021b. Synthetic Temporal Anomaly Guided End-To-End Video Anomaly Detection, in: Proceedings of ICCVW, pp. 207–214. Ba, J., Caruana, R., 2014. Do deep nets really need to be deep?, in: Proceedings of NIPS, pp. 2654–2662. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C., 2020. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings, in: Proceedings of CVPR, pp. 4183–4192. Bărbăluă, A., Ionescu, R.T., Georgescu, M.I., Duelohm, J., Ramachandra, B., Nasrollahi, K., Khan, F.S., Moeslund, T.B., Shah, M., 2023. SSMTL++: Revisiting Self-Supervised Multi-Task Learning for Video Anomaly Detection. Computer Vision and Image Understanding 229, 103656. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers, in: Proceedings of ECCV, pp. 215–229. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y., 2021. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. arXiv preprint arXiv:2102.04306 Chen, L., You, Z., Zhang, N., Xi, J., Le, X., 2022. UTRAD: Anomaly detection and localization with U-Transformer. Neural Networks 147, 53–62. Cheng, H., Yang, L., Liu, Z., 2021. Relation-based knowledge distillation for anomaly detection, in: Proceedings of PRCV, pp. 105–116. Cheng, K.W., Chen, Y.T., Fang, W.H., 2015. Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression, in: Proceedings of CVPR, pp. 2909–2917. Chung, I., Park, S., Kim, J., Kwak, N., 2020. Feature-map-level online adversarial knowledge distillation, in: Proceedings of ICMIL, pp. 2006–2015. Cong, Y., Yuan, J., Liu, J., 2011. Sparse reconstruction cost for abnormal event detection, in: Proceedings of CVPR, pp. 3449–3456. Del Giorno, A., Bagnell, J., Hebert, M., 2016. A Discriminative Framework for Anomaly Detection in Large Videos, in: Proceedings of ECCV, pp. 334–349. Deng, H., Li, X., 2022. Anomaly detection via reverse distillation from oneclass embedding, in: Proceedings of CVPR, pp. 9737–9746. Doshi, K., Yilmaz, Y., 2020. Any-Shot Sequential Anomaly Detection in Surveillance Videos, in: Proceedings of CVPRW, pp. 934–935. Doshi, K., Yilmaz, Y., 2021. An efficient approach for anomaly detection in traffic videos, in: Proceedings of CVPR, pp. 4236–4244. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: Proceedings of ICLR. Dutta, J.K., Banerjee, B., 2015. Online Detection of Abnormal Events Using Incremental Coding Length, in: Proceedings of AAAI, pp. 3755–3761. Fang, Z., Zhou, J.T., Xiao, Y., Li, Y., Yang, F., 2020. Multi-encoder towards effective anomaly detection in videos. IEEE Transactions on Multimedia 23, 4106–4116.

Fei, Y., Huang, C., Jinkun, C., Li, M., Zhang, Y., Lu, C., 2022. Attribute Restoration Framework for Anomaly Detection. IEEE Transactions on Multimedia 24, 116–127. Feng, Y., Wang, H., Hu, R., Yi, D.T., 2020. Triplet distillation for deep face recognition, in: Proceedings of ICIP, pp. 808–812. Feng, Y., Yuan, Y., Lu, X., 2017. Learning deep event models for crowd anomaly detection. Neurocomputing 219, 548–556. Georgescu, M.I., Bărbăluă, A., Ionescu, R.T., Khan, F.S., Popescu, M., Shah, M., 2021. Anomaly Detection in Video via Self-Supervised and Multi-Task Learning, in: Proceedings of CVPR, pp. 12742–12752. Georgescu, M.I., Dužǎ, G.E., Ionescu, R.T., 2022a. Teacher–student training and triplet loss to reduce the effect of drastic face occlusion: Application to emotion recognition, gender identification and age estimation. Machine Vision and Applications 33, 12. Georgescu, M.I., Ionescu, R., Khan, F.S., Popescu, M., Shah, M., 2022b. A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, 4505–4523. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Van Den Hengel, A., 2019. Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection, in: Proceedings of ICCV, pp. 1705–1714. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Proceedings of NIPS, pp. 2672–2680. Jou, G., Yu, B., Maybank, S.J., Tao, D., 2021. Knowledge distillation: A survey. International Journal of Computer Vision 129, 1789–1819. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S., 2016. Learning temporal regularity in video sequences, in: Proceedings of CVPR, pp. 733–742. Hinami, R., Mei, T., Satoh, S., 2017. Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge, in: Proceedings of ICCV, pp. 3639–3647. Hinton, G., Vinyals, O., Dean, J., 2014. Distilling the Knowledge in a Neural Network, in: Proceedings of NIPS Deep Learning and Representation Learning Workshop. Ionescu, R.T., Khan, F.S., Georgescu, M.I., Shao, L., 2019a. Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video, in: Proceedings of CVPR, pp. 7842–7851. Ionescu, R.T., Smeureanu, S., Alexe, B., Popescu, M., 2017. Unmasking the abnormal events in video, in: Proceedings of ICCV, pp. 2895–2903. Ionescu, R.T., Smeureanu, S., Popescu, M., Alexe, B., 2019b. Detecting abnormal events in video using Narrowed Normality Clusters, in: Proceedings of WACV, pp. 1951–1960. Jin, P., Mou, L., Xia, G.S., Zhu, X.X., 2022. Anomaly detection in aerial videos with transformers. IEEE Transactions on Geoscience and Remote Sensing 60, 1–13. Kim, J., Grauman, K., 2009. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates, in: Proceedings of CVPR, pp. 2921–2928. Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: Proceedings of ICLR. Lee, S., Kim, H.G., Ro, Y.M., 2019. BMAN: Bidirectional Multi-Scale Aggregation Networks for Abnormal Event Detection. IEEE Transactions on Image Processing 29, 2395–2408. Li, B., Leroux, S., Simoens, P., 2021. Decoupled appearance and motion learning for efficient anomaly detection in surveillance video. Computer Vision and Image Understanding 210, 103249. Li, G., Cai, G., Zeng, X., Zhao, R., 2022. Scale-Aware Spatio-Temporal Relation Learning for Video Anomaly Detection, in: Proceedings of ECCV, pp. 333–350. Liu, W., Luo, W., Lian, D., Gao, S., 2018a. Future Frame Prediction for Anomaly Detection – A New Baseline, in: Proceedings of CVPR, pp. 6536–6545. Liu, Y., Li, C.L., Pizcros, B., 2018b. Classifier Two-Sample Test for Video Anomaly Detections, in: Proceedings of BMVC. Liu, Z., Nie, Y., Long, C., Zhang, Q., Li, G., 2021. A Hybrid Video Anomaly Detection Framework via Memory-Augmented Flow Reconstruction and Flow-Guided Frame Prediction, in: Proceedings of ICCV, pp. 13588–13597. Lu, C., Shi, J., Jia, J., 2013. Abnormal Event Detection at 150 FPS in MAT-LAB, in: Proceedings of ICCV, pp. 2720–2727. Lu, Y., Yu, F., Kumar, M., Reddy, K., Wang, Y., 2020. Few-Shot Scene-
