CLIP-TSA: CLIP-Assisted Temporal Self-Attention for Weakly-Supervised Video Anomaly Detection

Hyekang Kevin Joo
Department of Computer Science
University of Maryland
College Park, MD

Khoa Vo
Department of Computer Science & Computer Engineering
University of Arkansas
Fayetteville, AR

David W. Jacobs
Department of Computer Science
University of Maryland
College Park, MD

Ngan Le
Department of Computer Science & Computer Engineering
University of Arkansas
Fayetteville, AR

Abstract

Video anomaly detection (VAD) – commonly formulated as a multiple-instance learning problem in a weakly-supervised manner due to its labor-intensive nature – is a challenging problem in video surveillance where the frames of anomaly need to be localized in an untrimmed video. In this paper, we first propose to utilize the ViT-encoded visual features from CLIP, in contrast with the conventional C3D or I3D features in the domain, to efficiently extract discriminative representations in the novel technique. We then model long- and short-range temporal dependencies and nominate the snippets of interest by leveraging our proposed Temporal Self-Attention (TSA). The ablation study conducted on each component confirms its effectiveness in the problem, and the extensive experiments show that our proposed CLIP-TSA outperforms the existing state-of-the-art (SOTA) methods by a large margin on two commonly-used benchmark datasets in the VAD problem (UCF-Crime and ShanghaiTech Campus). The source code will be made publicly available upon acceptance.

1 Introduction

Video understanding is a growing field and a subject of intense research that requires analysis of both spatial and temporal information, e.g., action recognition (Pareek & Thakkar, 2021; Vu et al., 2021a,b; Sun et al., 2022; Vu et al., 2022), action detection (Xu et al., 2020; Zeng et al., 2019; Vo et al., 2021a; Zhang et al., 2022), video captioning (Lei et al., 2020a; Dai et al., 2019; Yamazaki et al., 2022), video retrieval (Snoek et al., 2009; Gabeur et al., 2020; Wang et al., 2021; Wray et al., 2021). One of the challenging problems in video understanding is video anomaly detection (VAD), which is the task of localizing anomalous events in a given video. VAD is an area of research that has several years of history, and it has been gaining more attraction in recent years (Hasan et al., 2016; Sultani et al., 2018; Wu & Liu, 2021). Generally, there are three main paradigms in VAD, namely, fully-supervised (Liu & Ma, 2019), unsupervised (Gong et al., 2019; Zaheer et al., 2022), and weakly-supervised (Thakare et al., 2022; Sultani et al., 2018).
Table 1: Comparison among multiple VAD approaches.

<table>
<thead>
<tr>
<th>Supervision</th>
<th>Normal</th>
<th>Abnormal</th>
<th>Annotation</th>
<th>Approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully-Supervised</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Liu &amp; Ma (2019)</td>
</tr>
<tr>
<td>Weakly-Supervised</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Sultani et al. (2018); Thakare et al. (2022); Purwanto et al. (2021); Tian et al. (2021); Zaheer et al. (2020); Sapkota &amp; Yu (2022)</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>Hasan et al. (2016); Gao et al. (2021); Wang &amp; Cherian (2019); Lu et al. (2013); Zaheer et al. (2022); Wu &amp; Liu (2021)</td>
</tr>
</tbody>
</table>

While it generally yields high performance, the supervised VAD requires fine-grained anomaly labels (i.e., frame-level normal/abnormal annotations in the training data). However, the problem has traditionally been difficult to solve in a fully supervised manner due to the labor-expensive nature of data collection. In general, anomaly detection annotation requires the annotator to localize and label anomalies in a video, or a large set of sequential frames. Unfortunately, this is a very strenuous labor for the annotator because, as anomalies can happen at any moment, almost all of the frames need to be observed carefully, leading to massive time consumption. Because of its time-consuming and labor-intensive nature, collecting a fully-annotated large-scale dataset is a difficult task for the supervised VAD. In unsupervised VAD learning, one-class classification (OCC) problem (Zaheer et al., 2020) is a common approach, in which the model is trained on only normal class samples with the assumption that unseen abnormal videos have high reconstruction errors. However, the performance of unsupervised VAD is usually poor because of its lack of prior knowledge of abnormality as well as its inability to capture all normality variations (Chandola et al., 2009). Compared to both unsupervised and supervised VAD, the weakly-supervised VAD is considered the most practical approach by many for VAD because of its competitive performance and annotation efficiency by employing video-level labels to reduce the cost of manual fine-grained annotations (Zaheer et al., 2020; Zhong et al., 2019). The comparison among various VAD approaches is shown in Table 1.

In the weakly-supervised VAD task, there exist two fundamental problems. First, anomalous-labeled frames tend to be dominated by normal-labeled frames, as the videos are untrimmed and there is no strict length requirement for the anomalies in the video. Second, the anomaly may not necessarily stand out against normality. As a result, it occasionally becomes challenging to localize anomaly snippets. In order to combat the issues, Sultani et al. (2018); Tian et al. (2021); Wu et al. (2020); Zhang et al. (2019); Zhu & Newsam (2019) have attempted to tackle the problem in multiple instance learning (MIL) frameworks, which treat a video as a bag containing multiple instances, each instance being a video snippet. A video is labeled as anomalous if any of its snippets are anomalous, and normal if all of its snippets are normal. Following the MIL framework, anomalous-labeled videos belong to the positive bag and normal-labeled videos belong to the negative bag.

Furthermore, the existing approaches encode the extracted visual content by applying a backbone, e.g., C3D (Ji et al., 2013), I3D (Carreira & Zisserman, 2017), 2Stream (Simonyan & Zisserman, 2014), which are pre-trained on action recognition tasks. Different from the action recognition problem, VAD depends on discriminative representations that clearly represent the events in a scene. Thus, those existing backbones, C3D, I3D, and 2Stream, are not suitable because of the domain gap (Liu & Ma, 2019). To address such limitation, we leverage the success of the recent "vision-language” works (Patashnik et al., 2021; Yang & Zou, 2022; Vo et al., 2022; Yamazaki et al., 2023), which have proved the effectiveness of feature representation learned via Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021). CLIP consists of two networks, a vision encoder and a text encoder, which are trained on 400 million text-image pairs collected from a variety of publicly available sources on the Internet. Given a set of words and an image, CLIP can estimate the semantic similarity between them. We thus leverage CLIP as a visual feature extractor. Furthermore, the existing MIL-based weakly-supervised VAD approaches are limited in dealing with an arbitrary number of abnormal snippets in an abnormal video. To address such an issue, we are inspired by the differentiable top-K operator (Cordonnier et al., 2021) and introduce a novel technique, termed top-$\kappa$ function, that localizes $\kappa$ snippets of interest in the video with differentiable hard attention in the similar MIL setting to demonstrate its effectiveness and applicability to the traditional, popular setting. Furthermore, we introduce the Temporal Self-Attention (TSA) Mechanism, which aims to generate the reweighed attention feature by measuring the abnormal degree of snippets. Our proposed CLIP-TSA follows the MIL framework and consists of three components corresponding to (i) Feature
Encoding by CLIP; (ii) Modeling snippet coherency in the temporal dimension with our Temporal Self-Attention and (iii) Localizing anomalous snippets with Difference Maximization Trainer. As the real-world anomalies are diverse, in order to show the applicability of our proposed method to multiple environments, we run experiments on three different datasets commonly used for the VAD evaluation: UCF-Crime (Sultani et al., 2018), ShanghaiTech Campus (Liu et al., 2018), and XD-Violence (Wu et al., 2020). In addition, we conduct an ablation study on the effectiveness of our proposed method. Throughout the paper, the term abnormal and anomaly will be used interchangeably.

Our contributions are summarized as follows:

- We propose a Temporal Self-Attention (TSA) mechanism that is applicable to the Weakly-Supervised VAD problems and acquires anomaly likelihood scores for video snippets.
- We leverage CLIP, which uses a ViT as a backbone for visual features, to introduce 1) novel usage of CLIP features and 2) novel type of contextual representation in analyzing videos consisting of abnormal actions.
- We empirically validate the usefulness of our proposed method by showing that, to the best of our knowledge, it achieves superior performance to all of the current SOTA methods benchmarked on UCF-Crime and ShanghaiTech Campus datasets under any type of supervision setting. As for the XD dataset, it beats the performance of all the SOTAs trained without auditory features for a fair comparison.

2 Related Work

2.1 Unsupervised VAD

Unsupervised anomaly detection approaches do not require labeled data during training. In such approaches, the usual patterns with only normal training samples are first encoded and distinctive encoded patterns are detected as anomalies. While the early anomaly detection methods (Anti´c & Ommer, 2011; Basharat et al., 2008; Li et al., 2013; Saligrama & Chen, 2012; Wu et al., 2019) mainly depend on the handcrafted features, the recent approaches primarily make use of the merits of deep neural networks (DNNs) (Doshi & Yilmaz, 2020; Hasan et al., 2016; Jonescu et al., 2019; Lu et al., 2013; Ramachandra et al., 2020; Wang & Cherian, 2019; Zaheer et al., 2022). In such approaches, reconstruction error is utilized to identify anomalies with the assumption that anomalous events are often reconstructed poorly. For example, Hasan et al. (2016) used autoencoders as feature extractors to model the subsequent frame and estimated abnormality by reconstruction error. Later, Wang & Cherian (2019) assumed that anomalous events will cause a big difference between past and future frames and proposed spatiotemporal autoencoder with combinations of CNNs and LSTMs (Hochreiter & Schmidhuber, 1997). With a similar assumption on reconstruction errors as an abnormality recognizer, Feng et al. (2021); Liu et al. (2018); Park et al. (2020) adopted generative networks to synthesize or predict future frames. Furthermore, Doshi & Yilmaz (2020) proposed a hybrid use of DNNs and statistical kNN (k nearest neighbor) decision approach for finding video anomalies. Siamese network was employed to detect anomaly (Ramachandra et al., 2020) by learning a distance function between a pair of video patches.

Historically, the performance of unsupervised anomaly detection problems generally lagged behind that of weakly-supervised anomaly detection by a large margin because the model in an unsupervised setting significantly lacks the prior knowledge of anomaly needed for differentiation between normality and anomaly.

2.2 Weakly-supervised VAD

Weakly-supervised VAD methods (Lv et al., 2021; Purwanto et al., 2021; Sapkota & Yu, 2022; Sultani et al., 2018; Thakare et al., 2022; Tian et al., 2021; Wu et al., 2020; Zaheer et al., 2020; Zhang et al., 2019; Zhong et al., 2019; Zhu & Newsam, 2019) rely on the video-level labels. In this setup, a normal-labeled video contains all normal events, whereas an anomaly-labeled video contains both normal and anomalous events without any temporal information about starting and ending of anomalous events. Weakly-supervised VAD problem has been generally regarded as an MIL problem (He et al., 2017; Huo et al., 2012; Sultani et al., 2018) as the videos are labeled at bag-level (i.e., video-level), with the anomaly-labeled video regarded as a positive bag and the normal-labeled video regarded as a negative bag. Particularly since Sultani et al. (2018) proposed a weakly-supervised framework to detect anomalies on UCF-Crime, in which both normal and abnormal samples annotated at video-level are included in both train and test sets, this research in the weakly-supervised setting has grown and gained significant popularity. Since then, more weakly-labeled VAD datasets, primarily for use in a weakly-supervised setting, have been introduced (Liu et al., 2018).
In such approaches, the feature extractor can be trained or utilized by pre-trained models. While Zhong et al. (2019), Zhu & Newsam (2019) trained both the feature encoder and classifier simultaneously, Sultani et al. (2018); Tian et al. (2021); Zhang et al. (2019) utilized pre-trained models such as C3D (Ji et al., 2013), 15D (Carreira & Zisserman, 2017), 2Stream (Simonyan & Zisserman, 2014), and SlowFast (Feichtenhofer et al., 2019) as feature extractors and trained the classifier only.

2.3 Vision-Language Pre-trained Models

Vision-language pre-trained model (VLPM) aims to learn the semantic correspondence between different modalities (i.e., video and text) by pre-training the model on a large-scale dataset of video/image-text pairs. Specifically, the model mines the associations between objects or actions in the video and objects or actions in the text. Standard vision-language tasks include video captioning (Krishna et al., 2017; Pasunuru & Bansal, 2017; Vo et al., 2022), text-to-video retrieval (Hendricks et al., 2018; Rohrbach et al., 2015), and video question answering (Girdhar & Ramanan, 2020; Lei et al., 2020b). Generally, VLPM can be divided into two categories: single-stream and dual-stream. The former uses a single transformer to model both image/video and text representations in a unified framework. Both image/video and text embeddings are concatenated into one feature. This category includes VisualBERT (Li et al., 2019), UNIMO (Li et al., 2020b), OSCAR (Li et al., 2020c), UNICODER (Li et al., 2020a), and UNITER (Chen et al., 2020b). The latter one separately encodes image/video and text with a decoupled encoder. This category includes LXMERT (Tan & Bansal, 2019), ViLBERT (Lu et al., 2019), CLIP (Radford et al., 2021), and DeCLIP (Li et al., 2021). VisualBERT, ViLBERT, OSCAR, UNICODER, UNITER, and LXMERT use masked token tasks and are based on Language Modeling, whereas UNIMO, CLIP, and DeCLIP are trained on contrastive learning. Because of simplicity, flexibility, and low computation cost, we adopt the frozen self-supervised vision-language model CLIP, a dual-stream architecture and contrastive learning in this paper.

2.4 Attention Mechanism

Attention models have a long history. In 2015, Bahdanau et al. (2015) introduced one of the first soft attention models capable of attending to all the source words and attempted to solve the machine translation task without the traditional encoder-decoder models (e.g., RNN, LSTM), which were common approach for the problem at the time (Cho et al., 2014; Sutskever et al., 2014). Shortly afterward, Xu et al. (2015) introduced a hard stochastic attention mechanism that is able to compute the relative importance of the source words with respect to the output words, combating the huge expense of computation required for soft attention. Because hard attention only places attention locally, the mechanism is generally computationally less expensive than the soft attention mechanism, which observes all hidden states (Luong et al., 2015). In general, while soft attention models are trainable end-to-end, hard attention models are not differentiable and require reinforcement learning (Xu et al., 2015). Today, many variations of attention mechanisms have been introduced. For example, Luong et al. (2015) proposed a local attention mechanism similar to hard attention, but is differentiable. In 2017, Vaswani et al. (2017) introduced a neural machine translation (NMT) architecture named Transformer that is designed with only fully connected layers and attention by leveraging the self-attention mechanism. Recently, Vo et al. (2021b, 2022) inherited the merits from both soft attention models and hard attention models and proposed adaptive attention models. Despite its original application in NMT, Transformer has been gaining great attraction, and its usage has expanded widely, including computer vision.

3 Proposed Method

3.1 Problem Setup

In weakly-supervised VAD, videos in the training set are only labeled at video-level. Let there be a set of weakly-labeled training videos \( S = \{ x^{(k)}_i, y^{(k)} \}_{k=1}^{|S|} \), where a video \( x^{(k)} \in \mathbb{R}^{N_k \times W \times H} \) is a sequence of \( N_k \) frames that are \( W \) pixels wide and \( H \) pixels high, and \( y^{(k)} = \{0, 1\} \) is the video-level label of video \( x^{(k)} \) in terms of anomaly (i.e., 1 if the video contains anomaly; 0 otherwise).

Given a video \( x^{(k)} \in \mathbb{R}^{N_k \times W \times H} \) consisting of \( N_k \) frames, i.e., \( x^{(k)} = \{ x_j \}_{j=1}^{N_k} \), we first divide \( x^{(k)} \) into a set of \( \delta \)-frame snippets \( \{ s_i \}_{i=1}^{N_k} \). Feature representation of each snippet is extracted by applying a vision-language model into the middle frame. In this work, CLIP is chosen as a vision-language model; however, it can be substituted by any
Figure 1: Overall flowchart of our proposed CLIP-TSA in train time. Given a video $X$ consisting of $N$ frames (i.e., $X = \{x_j\}_{j=1}^N$), we first divide into a set of $\delta$-frame snippets $\{s_i\}_{i=1}^T$. Each $\delta$-frame snippet $s_i$ is represented by a vision-language feature $f_i \in \mathbb{R}^d$. Then, the features $F = \{f_i\}_{i=1}^T$, where $f_i \in \mathbb{R}^d$, are resized into one uniform length $T$ to allow batch training by following Eq. 1. Our proposed TSA is then applied onto the resized features to obtain anomaly attention feature $\hat{F} = \{\hat{f}_i\}_{i=1}^T$, where $\hat{f}_i \in \mathbb{R}^d$. The anomaly attention feature $\hat{F}$ is used for: 1) producing an anomaly likelihood score $U$ using the score classifier $C$; 2) optimizing the model by employing the difference maximization trainer technique $\nu_{\gamma,\alpha}$ using the feature magnitude.

vision-language model as introduced in Section 2.3. Thus, each $\delta$-frame snippet $s_i$ is represented by a vision-language feature $f_i \in \mathbb{R}^d$ and the video $X^{(k)}$ is represented by a set of video feature vectors $F_k = \{f_i\}_{i=1}^{T_k}$, where $f_i \in \mathbb{R}^d$ and $T_k$ is the number snippets of $X^{(k)}$.

CLIP-TSA is trained using a mini-batch; thus, it introduces an issue caused by the difference in video embedding feature length $T$ between samples in the mini-batch. To address this issue, we normalize video feature length by following the approach introduced by (Sultani et al., 2018). Given two videos $X^{(1)}$ and $X^{(2)}$, their corresponding sets of video feature vectors are $F_1 = \{f_i\}_{i=1}^{T_1}$ and $F_2 = \{f_i\}_{i=1}^{T_2}$, respectively, where $T_1 \neq T_2$. Following their paradigm, both $F_1$ and $F_2$ with size $T_1$ and $T_2$ are reshaped into the same size of $T$ with Eq. 1 where $[g] = \left\lfloor \frac{T}{T_1} \right\rfloor$ and $[g'] = \left\lfloor \frac{T}{T_2} \right\rfloor$ for videos $X^{(1)}$ and $X^{(2)}$, respectively:

$$F = \{f_i'\}_{i'=1}^{T} = \frac{1}{[g]} \sum_{i=g \cdot (i'-1)}^{g \cdot i'} f_i$$

Using this technique, we can handle an arbitrary length of videos, allowing for training the features in batches. However, in test time, as the videos are evaluated one at a time, the features do not go through the normalization process in test time. In this paper, we assume that, in training time, the input features $F$ come post-normalized into the uniform shape in temporal dimension $T$ for batch training.

Our proposed anomaly detection CLIP-TSA’s pipeline is portrayed in Figure 1 with three main components i.e., (i) Feature Encoding, (ii) Temporal Self Attention (TSA), and (iii) Difference Maximization, which are elaborated in the following sections.

### 3.2 Feature Encoding

CLIP (Radford et al., 2021) is an image-text matching model, and it has recently attained remarkable achievements in various computer vision tasks such as image classification (Cheng et al., 2021), image-text retrieval (Dzabraev et al., 2021), and image generation (Patashnik et al., 2021). Originally, CLIP is trained to match an image with its corresponding natural language descriptions. CLIP consists of two independent encoders respectively for visual and textual features encoding. Given a batch of images and texts, CLIP aims to align their feature in the embedding space with a contrastive loss during the training process. CLIP is comprehensively trained on 400 million image-text pairs.
3.3 Temporal Self-Attention (TSA)

Our proposed TSA mechanism aims to model the coherency between snippets of a video and select the top-$\kappa$ most relevant snippets. It contains three modules i.e., (i) temporal scorer network, (ii) top-$\kappa$ score nominator, and (iii) fusion network, as visualized in Figure 2 and mathematically explained in Algorithm 1.

In TSA, the vision language feature $\mathcal{F} \in \mathbb{R}^{T \times d}$ (from 3.2 Feature Encoding) is first converted into a score vector $\omega \in \mathbb{R}^T$ through a temporal scorer network $\phi_s$, i.e., $\omega = \phi_s(\mathcal{F})$. This network is meant to be shallow; thus, we choose a multi-layer perceptron (MLP) of 3 layers in this paper. The scores, each of which is representing the snippet $s_i$, are then passed into the top-$\kappa$ score nominator to extract the $\kappa$ most relevant snippets from the video. The top-$\kappa$ score nominator is implemented by the following two steps. First, the scores $\omega$ are cloned $M$ times and the cloned score $\bar{\omega} \in \mathbb{R}^{T \times M}$ is obtained; $M$ represents the number of independent samples of score vector $\omega$ to generate for the empirical mean, which is to be used later for computing the expectation with noise-perturbed features.
Throughout the paper, we set $M$ to be 100. Second, Gaussian noise $G \in \mathbb{R}^{T \times M}$ is applied to the stack of $M$ clones by the following Eq. 3 to produce $\tilde{\omega} \in \mathbb{R}^{T \times M}$:

$$\tilde{\omega} = G \oplus \omega$$

where $\oplus$ is an element-wise addition (3)

From the Gaussian-perturbed scores $\tilde{\omega} \in \mathbb{R}^{T \times M}$, the indices of top-$\kappa$ snippets are selected based on the score magnitude independently across its $M$ dimension to represent the most relevant snippets and are later one-hot encoded into a matrix $V = \{V_i\}_{i=1}^{M}$, with each $V_i \in \mathbb{R}^{\kappa \times T}$ containing a set of one-hot vectors. More specifically, we guide the network to place the attention on $\kappa$ magnitudes with the highest values because the Difference Maximization Trainer (Sec 3.4) trains the anomalous snippets to have a high value and the normal snippets to have a low value. The matrix $V$ is then averaged across its $M$ dimension to produce a stack of soft one-hot vectors $\hat{V} \in \mathbb{R}^{\kappa \times T}$. Through the soft one-hot encoding mechanism, the higher amount of attention, or weight, is placed near and at the indices of top-$\kappa$ scores (e.g., $[0, 0, 1, 0] \rightarrow [0, 0.03, 0.95, 0.02]$). The top-$\kappa$ score nominator can be summarized by the pseudocode in Algorithm 1.

Afterwards, the stack of perturbed soft one-hot vectors $\hat{V} \in \mathbb{R}^{\kappa \times T}$ is transformed into $\hat{V} \in \mathbb{R}^{\kappa \times T \times d}$ by making $d$ clones of $\hat{V}$, and the set of input feature vectors $F \in \mathbb{R}^{T \times d}$ is transformed into $\hat{F} \in \mathbb{R}^{\kappa \times T \times d}$ by making $\kappa$ clones of $F$. Next, the matrices $\hat{V}$ and $\hat{F}$, which carry the reweighed information of snippets and represent the input video features, respectively, are fused together to create a perturbed feature $Q \in \mathbb{R}^{\kappa \times T \times d}$ that represents the reweighed feature magnitudes of snippets based on the previous computations as follows:

$$Q = \hat{V} \otimes \hat{F}$$

where $\otimes$ is an element-wise multiplication (4)

Then, each stack of perturbed feature vectors $Q \in \mathbb{R}^{\kappa \times d}$ within the perturbed feature $Q = \{Q_i\}_{i=1}^{T}$ is independently summed up across its dimension $\kappa$ to combine the magnitude information of $Q_i$ into one vector $f_i \in \mathbb{R}^d$. This step is akin to the process of reversing the previous one-hot encoding procedure by reducing the dimension dimension previously expanded for one-hot encoding. The reweighed feature vector, $f_i \in \mathbb{R}^d$, which collectively forms $\tilde{F} = \{f_i\}_{i=1}^{T}$, is collectively obtained as the model output from the TSA mechanism $\sigma$ to represent an anomaly attention feature $\tilde{F} \in \mathbb{R}^{T \times d}$. The pipeline of TSA is described by the pseudocode in Algorithm 1 and illustrated in Figure 2.

### Algorithm 1: TSA mechanism $\sigma$ to produce anomaly attention features $\tilde{F}$

**Data:** Feature $F \in \mathbb{R}^{T \times d}$, Top snippet count $\kappa \in \mathbb{R}^1$

**Result:** Anomaly attention feature $\tilde{F}$

1. $\omega \leftarrow \phi_x(F)$ \hspace{1cm} // $\mathbb{R}^{T \times 1}$
2. $\hat{Y} \leftarrow \text{Top-} \kappa \text{ Score}(M, \kappa, \omega)$ \hspace{1cm} // Alg 2, $\mathbb{R}^{\kappa \times T}$
3. $\hat{Y} \leftarrow \text{Make } d \text{ clones of } \hat{Y}$ \hspace{1cm} // $\mathbb{R}^{\kappa \times T \times d}$
4. $\hat{F} \leftarrow \text{Make } \kappa \text{ clones of } \hat{Y}$ \hspace{1cm} // $\mathbb{R}^{\kappa \times T \times d}$
5. $Q \leftarrow \hat{Y} \otimes \hat{F}$ \hspace{1cm} // $\mathbb{R}^{\kappa \times T \times d}$
6. $\tilde{F} \leftarrow \text{summation of } Q \text{ across dim } \kappa$ \hspace{1cm} // $\mathbb{R}^{T \times d}$
7. return $\tilde{F}$ \hspace{1cm} // dim:dimension

### Algorithm 2: Top-$\kappa$ Score function

**Data:** Sample count $M$, Top snippet count $\kappa$, Score vector $\omega$

**Result:** A stack of soft one-hot vectors $\hat{Y}$

1. set $\omega$ to $M$ clones of $\omega$ \hspace{1cm} // $\mathbb{R}^{T \times M}$
2. set $G$ to Gaussian noise \hspace{1cm} // $\mathbb{R}^{T \times M}$
3. $\tilde{\omega} \leftarrow G \oplus \omega$ \hspace{1cm} // $\mathbb{R}^{T \times M}$
4. $U \leftarrow \text{indices of top-} \kappa \text{ scores}$ \hspace{1cm} // $\mathbb{R}^{\kappa \times M}$
5. $\hat{Y} \leftarrow \text{one-hot encode } \kappa \text{ in } U$ \hspace{1cm} // $\mathbb{R}^{\kappa \times T \times M}$
6. $\hat{Y} \leftarrow \text{average of } \hat{Y} \text{ across dim } M$ \hspace{1cm} // $\mathbb{R}^{\kappa \times T}$
7. return $\hat{Y}$ \hspace{1cm} // dim:dimension

### 3.4 Difference Maximization Trainer Learning

Our weakly-supervised VAD model, CLIP-TSA, is set up as an MIL framework, in which the positive bag represents anomaly and the negative bag denotes normality. Following the paradigm, a video, treated as a bag, is labeled a positive bag if it contains at least one snippet of anomaly, while it is labeled a negative bag otherwise. Given a mini-batch of $2 \times B$ videos $\{X^{(k)}\}_{k=1}^{2B}$, each video $X^{(k)}$ is represented by $F_k = \{f_i\}_{i=1}^{T}$ obtained by TSA (Section 3.3). Let the input mini-batch be represented by $Z = \{F_k\}_{k=1}^{2B} \in \mathbb{R}^{2B \times T \times d}$, where $B$, $T$, and $d$ denote the user-input batch size, normalized time snippet count, and feature dimension, respectively. The actual batch size is dependent on the user-input batch size, following the equation of $2 \times B$, because the first half, $Z_- \in \mathbb{R}^{B \times T \times d}$, is loaded with a set of normal bags, and the second half, $Z_+ \in \mathbb{R}^{B \times T \times d}$, is loaded with a set of abnormal bags in order within the mini-batch.
After the mini-batch undergoes the phase of TSA, it outputs a set of reweighed normal attention features \( \hat{Z}_- = \{ \hat{F}_k \}_{k=1} \) and a set of reweighed anomaly attention features \( \hat{Z}_+ = \{ \hat{F}_k \}_{k=1} \). The reweighed attention features \( \hat{Z} \) are then passed into a convolutional network module \( J \) composed of dilated convolutions (Yu & Koltun, 2016) and non-local block (Wang et al., 2018) to model the long- and short-term relationship between snippets based on the reweighed magnitudes. The resulting stack of convoluted attention features \( \tilde{Z} = \{ \tilde{F}_k \}_{k=1} \), where \( \tilde{Z} \in \mathbb{R}^{2\times B \times T \times d} \), is then passed into a shallow MLP-based score classifier network \( C \) that converts the features into a set of scores \( U \in \mathbb{R}^{2\times B \times T \times d} \) to determine the binary anomaly state of feature snippets. The set of scores \( U \) is saved as part of a group of returned variables, for use in loss.

Next, each convoluted attention feature \( \{ \hat{F}_k \}_{k=1} \) of the batch \( \hat{Z} \) undergoes Difference Maximization Trainer (DMT). Leveraging the top-\( \alpha \) instance separation idea employed by [Li & Vasconcelos, 2015; Sultani et al., 2018], we use DMT, represented by \( v_{\gamma,\alpha} \), in this problem to maximize the separation, or difference, between top instances of two contrasting bags, \( \hat{Z}_- \) and \( \hat{Z}_+ \), by first picking out the top-\( \alpha \) snippets from each convoluted attention feature \( \hat{F}_k \) based on the feature magnitude. This produces a top-\( \alpha \) subset \( \hat{F}_k \in \mathbb{R}^{n \times d} \) for each convoluted attention feature \( \hat{F}_k \in \mathbb{R}^{T \times d} \). Second, \( \hat{F}_k \) is averaged out across top-\( \alpha \) snippets to create one feature vector \( \hat{F}_k \in \mathbb{R}^d \) that represents the bag. The procedure is explained by Eq. 5 below:

\[
\lambda_{\gamma,\alpha}(\hat{F}) = \hat{F} = \max_{\Omega(\tilde{F}) = \{ \tilde{F}_i \}_{i=1}^T} \frac{1}{\alpha} \sum_{\tilde{F}_i \in \Omega(\tilde{F})} \tilde{f}_i
\]  

(5)

In the equation, \( \lambda \) is parameterized by \( \gamma \), which denotes its dependency on the ability of the convolutional network module \( J \) (i.e., representation of \( F \)) depends on the top-\( \alpha \) positive instances selected with respect to \( J \). In addition, \( \alpha \) in Eq. 5 denotes the size of \( \Omega \), where \( \Omega \) represents a subset of \( \alpha \) snippets from \( F \). Each representative vector \( F \) is then normalized to produce \( \tilde{F} \in \mathbb{R}^1 \).

\[
v_{\gamma,\alpha}(\hat{F}_+,\hat{F}_-) = ||\lambda_{\gamma,\alpha}(\hat{F}_+)|| - ||\lambda_{\gamma,\alpha}(\hat{F}_-)||
\]  

(6)

The separability is computed as in Eq. 6, where \( \hat{F}_- = \{ \hat{f}_i \}_{i=1}^T \) represents a negative bag and \( \hat{F}_+ = \{ \hat{f}_i \}_{i=1}^T \) represents a positive bag. More specifically, we leverage the theorem below to maximize the separability of the top-\( \alpha \) instances (feature snippets) from each contrasting bag.

**Theorem 1 [Li & Vasconcelos, 2015; Tian et al., 2021]**: Expected Separability. Let \( E[||\hat{f}_+||_2] \geq E[||\hat{f}_-||_2] \), where \( \hat{F}_+ \) has \( \epsilon \in [1,T] \) abnormal samples and \( (T-\epsilon) \) normal samples, \( \hat{F}_- \) has \( T \) normal samples, and \( T = |\hat{F}_+| = |\hat{F}_-| \). Let \( \Upsilon_{\gamma,\alpha}() \) be the random variable from which the separability scores \( v_{\gamma,\alpha}(\cdot) \) of Eq. 6 are drawn.

1. If \( 0 < \alpha < \epsilon \), then

\[
0 \leq E[\Upsilon_{\gamma,\alpha}(\hat{F}_+,\hat{F}_-)] \leq E[\Upsilon_{\gamma,\alpha+1}(\hat{F}_+,\hat{F}_-)]
\]  

(7)

2. For a finite \( \epsilon \), then

\[
\lim_{\alpha \to \infty} E[\Upsilon_{\gamma,\alpha}(\hat{F}_+,\hat{F}_-)] = 0
\]  

(8)

In simple terms, the theorem in our setting conveys that, as the number of samples in the top-\( \alpha \) snippets of the abnormal video increases – but no greater than \( \epsilon \) – the separability between the two contrasting bags may be maximized. However, if it exceeds the number, it becomes difficult as the number of negative (normal) samples starts to dominate in both negative and positive bags.

Afterward, to compute the loss, a batch of normalized representative features \( \{ \tilde{F}_{normal} \}_{k=1}^B \) and \( \{ \tilde{F}_{abnormal} \}_{k=1}^B \) are then measured for margins between each other. A batch of margins is then averaged out and used as part of the net loss together with the score-based binary cross-entropy loss computed using the score set \( U \).

### 3.5 Inference

In test time, the video feature vectors \( F \) that have been extracted with CLIP do not undergo the normalization process to be reshaped into the common size of \( T \) because each feature is evaluated at a time. When \( \hat{F}_k \in \mathbb{R}^{T_k \times d} \) is input into the model in test time, the feature \( \hat{F}_k \) undergoes the proposed TSA process to produce the reweighed attention features \( \tilde{F}_k \). They are then passed into the convolutional network module \( J \), followed by the MLP-based score classifier network \( C \), to acquire a set of scores \( U \in \mathbb{R}^{T_k \times 1} \). Each score \( \{ u_i \}_{i=1}^{T_k} \) within this set of scores \( U \) represents the anomaly
likelihood of the snippet at the corresponding index and carries a value between 0 and 1. Each score $u_i$ is rounded to produce a set of binary scores $U' = \{u'_i\}_{i=1}^{T_k}$. When the binary score $u'_i$ is 1, the snippet at the corresponding index is deemed to be anomalous; whereas, when the score is 0, the snippet at the corresponding index is assumed to be normal. Lastly, each binary score in $U'$ is repeated $\delta$ times, preserving the original order, to reproduce a vector $\hat{U} = \{\hat{u}_i\}_{i=1}^{\delta \times T_k}$ with the common frame length as the video $X^{(k)}$, for use in evaluation against the ground truth labels as in Eq. 9 below. The remainder frames $N_k - \delta \times T_k$ are either discarded or padded with the final label of the video.

$$\hat{u}_{[\delta i: \delta (i+1)]} = u'_i$$ (9)

### 4 Experimental Results

#### 4.1 Datasets and Metrics

**UCF-Crime Dataset** (Sultani et al., 2018) contains 1,900 untrimmed video clips encompassing 13 different anomalies and normal activities. The types of anomalies in the videos include abuse, arrest, arson, assault, burglary, explosion, fighting, road accident, robbery, shooting, shoplifting, stealing, and vandalism. Each of the real-world surveillance videos, totaling 128 hours in length, has been weakly annotated at video-level as anomalous or normal. The dataset comes pre-split into a train set of 800/810 normal/anomalous videos; a test set of 150/140 normal/anomalous videos.

**ShanghaiTech Campus Dataset** (Liu et al., 2018) contains 317,398 frames of video clips encompassing the scenes of multiple areas in ShanghaiTech Campus. The dataset cumulatively covers 13 scenes, in which 300,308 frames represent normal events and the remaining 17,090 frames comprise 130 distinct anomalous events. The dataset is split into a train set of 330 videos (274,515 frames) and a test set of 107 videos (42,883 frames), captured at 480×856 pixels. The train set contains only normal videos, while the test set contains a mix of normal and anomalous videos, where the anomalies in the test set are annotated at pixel-level.

**XD-Violence Dataset** (Wu et al., 2020) contains 217 hours of 4,754 untrimmed videos encompassing six different anomalies and normal activities. The anomalous actions in the dataset include abuse, car accident, explosion, fighting, riot, and shooting. The train set contains video-level annotations, while the test set contains frame-level annotations (i.e., rough from-and-to frame locations of each anomaly, not to exceed three, in a video). The dataset is split into a train set of 3,954 videos, where 1,905 of them are anomalous, and a test set of 800 videos, where 500 are anomalous.

**Metrics:** Similar to other work (Hasan et al., 2016; Sultani et al., 2018; Tian et al., 2021; Wu & Liu, 2021; Wu et al., 2020; Zhong et al., 2019), UCF-Crime and Shanghai datasets are evaluated using AUC@ROC and XD-Violence dataset is evaluated using AUC@PR. AUC@ROC refers to the area under the receiver operating characteristics curve, whereas AUC@PR refers to the area under the precision-recall curve.

#### 4.2 Implementation Details

In training time, we follow Sultani et al. (2018); Tian et al. (2021) and divide each video in the batch into 32 video snippets, (i.e., $T$ is set as 32 in train time), using Eq. 1. For all datasets, we follow the aforementioned steps to preprocess videos with the snippet length set to $\delta = 16$. The scorer network $\theta_s$ in Section 3.3 is defined as an MLP of three layers of 512, 256, and 1 units. The hidden layer is followed by a ReLU activation function, and the final layer is followed by a sigmoid function to produce a value between 0 and 1. To extract the linguistic scene elements features of the scene, we employ CLIP (Radford et al., 2021) that was pre-trained on a large-scale dataset of 400M image-text pairs crawled from the Internet. Thus, $d$ is set as 512 for all experiments. We set $M$ as 100 for Gaussian noise in Eq. 3. In addition, we choose 0.7 (70%), 0.7 (70%), and 0.9 (90%) for $r$ in UCF-Crime, ShanghaiTech Campus, and XD-Violence datasets, respectively, for the best performance, where $r$ denotes the number of snippets in a feature to place attention onto using TSA in a proportionate, relative figure rather to later compute $\kappa$ in a hard number:

$$\kappa = \lfloor T \times r \rfloor$$ (10)

Our CLIP-TSA is trained in an end-to-end manner and implemented using PyTorch. We use the Adam optimizer (Kingma & Ba, 2015) with a weight decay of 0.005 and a batch size of 16 for 4,000 (UCF-Crime), 35,000 (ShanghaiTech Campus), and 4,000 (XD-Violence) epochs. The learning rate is set to 0.001 for all datasets.
Table 2: Performance comparisons (AUC@ROC) between the SOTA methods and our method on UCF-Crime dataset (Sultani et al., 2018). They are grouped into the unsupervised, supervised, and weakly-supervised methods in order.

<table>
<thead>
<tr>
<th>Sup.</th>
<th>Method</th>
<th>Venue</th>
<th>Feature</th>
<th>AUC@ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un-</td>
<td>Hasan et al. (2016)</td>
<td>CVPR’16</td>
<td>-</td>
<td>50.60</td>
</tr>
<tr>
<td></td>
<td>Lu et al. (2013)</td>
<td>ICCV’13</td>
<td>C3D</td>
<td>65.51</td>
</tr>
<tr>
<td></td>
<td>BODS (Wang &amp; Cherian, 2019)</td>
<td>ICCV’19</td>
<td>I3D</td>
<td>68.26</td>
</tr>
<tr>
<td></td>
<td>GODS (Wang &amp; Cherian, 2019)</td>
<td>ICCV’19</td>
<td>I3D</td>
<td>70.46</td>
</tr>
<tr>
<td></td>
<td>GCL (Zaheer et al., 2019)</td>
<td>CVPR’22</td>
<td>ResNext</td>
<td>71.04</td>
</tr>
<tr>
<td>Fully-</td>
<td>Liu &amp; Ma (2019)</td>
<td>MM’19</td>
<td>NLN</td>
<td>82.0</td>
</tr>
<tr>
<td></td>
<td>GCL (Zaheer et al., 2022)</td>
<td>CVPR’21</td>
<td>ResNext</td>
<td>79.84</td>
</tr>
<tr>
<td></td>
<td>GCN (Zhong et al., 2019)</td>
<td>CVPR’19</td>
<td>TSN</td>
<td>82.12</td>
</tr>
<tr>
<td></td>
<td>WSAL (Lv et al., 2021)</td>
<td>TIP’21</td>
<td>TSN</td>
<td>85.38</td>
</tr>
<tr>
<td></td>
<td>Purwanto et al. (2021)</td>
<td>ICCV’21</td>
<td>TRN</td>
<td>85.00</td>
</tr>
<tr>
<td></td>
<td>Thakare et al. (2022)</td>
<td>ExpSys’22</td>
<td>C3D+I3D</td>
<td>84.48</td>
</tr>
<tr>
<td>Weakly-</td>
<td>Sultani et al. (2018)</td>
<td>CVPR’18</td>
<td></td>
<td>75.41</td>
</tr>
<tr>
<td></td>
<td>Zhang et al. (2019)</td>
<td>ICCP’19</td>
<td></td>
<td>78.70</td>
</tr>
<tr>
<td></td>
<td>GCN (Zhong et al., 2019)</td>
<td>CVPR’19</td>
<td>C3D</td>
<td>81.08</td>
</tr>
<tr>
<td></td>
<td>CLAWS (Zaheer et al., 2020)</td>
<td>ECCV’20</td>
<td></td>
<td>83.03</td>
</tr>
<tr>
<td></td>
<td>RTFM (Tian et al., 2021)</td>
<td>ICCV’21</td>
<td></td>
<td>83.28</td>
</tr>
<tr>
<td>Ours: CLIP-TSA†</td>
<td></td>
<td></td>
<td></td>
<td>83.94</td>
</tr>
<tr>
<td></td>
<td>Sultani et al. (2018)</td>
<td>CVPR’18</td>
<td></td>
<td>77.92</td>
</tr>
<tr>
<td></td>
<td>Wu et al. (2020)</td>
<td>ECCV’20</td>
<td></td>
<td>82.44</td>
</tr>
<tr>
<td></td>
<td>DAM (Majhi et al., 2021)</td>
<td>AVSS’21</td>
<td>I3D</td>
<td>82.67</td>
</tr>
<tr>
<td></td>
<td>BN-SVP (Sapkota &amp; Yu, 2022)</td>
<td>CVPR’22</td>
<td></td>
<td>83.39</td>
</tr>
<tr>
<td></td>
<td>RTFM (Tian et al., 2021)</td>
<td>ICCV’21</td>
<td></td>
<td>84.30</td>
</tr>
<tr>
<td></td>
<td>Wu &amp; Liu (2021)</td>
<td>TIP’21</td>
<td></td>
<td>84.89</td>
</tr>
<tr>
<td>Ours: CLIP-TSA†</td>
<td></td>
<td></td>
<td></td>
<td>84.66</td>
</tr>
<tr>
<td>Ours: CLIP-TSA ‡</td>
<td>CLIP</td>
<td></td>
<td></td>
<td>87.58</td>
</tr>
</tbody>
</table>

4.3 Performance Comparison

Besides CLIP-TSA, which is conducted on vision-language feature CLIP and our temporal attention mechanism TSA, we also test CLIP-TSA on other common features, i.e., C3D (Ji et al., 2013) and I3D (Carreira & Zisserman, 2017), to fairly compare CLIP-TSA with other existing approaches. Thus, we report the performance of the CLIP-TSA and its variants in this section as follows:

- **CLIP-TSA†**: replacement of CLIP feature with C3D (Ji et al., 2013) feature (TSA Preserved)
- **CLIP-TSA‡**: replacement of CLIP feature with I3D (Carreira & Zisserman, 2017) feature (TSA Preserved)
- **CLIP-TSA**: utilization of the CLIP feature and the temporal attention mechanism TSA

Table 2 shows the frame-level AUC@ROC results of SOTA models that we have found to the best of our ability on the UCF-Crime dataset. Based on the table, first, it is apparent that unsupervised methods generally provide an inferior performance. Second, it can be observed that the performance of our method, CLIP-TSA, stands out against other SOTA methods by a large margin in any type of supervision setting. Compared to the current best-performing model, i.e., [Lv et al., 2021], our CLIP-TSA holds 2.2% better performance when evaluated with the same metric. Furthermore, on the same feature, our CLIP-TSA† yields better performance than the current SOTA on C3D by 0.66%, and CLIP-TSA‡ obtains very competitive scores on I3D.

Similarly, Table 3 shows the frame-level AUC@ROC results of SOTA models on the ShanghaiTech Campus dataset. In the table, it can be seen that our model outperforms all of the previous SOTA methods reported in the table. Empirically, it shows that, on the same feature, CLIP-TSA† beats BN-SVP (Sapkota & Yu, 2022), the current SOTA on C3D, by 1.19%, and CLIP-TSA‡ outperforms Wu & Liu (2021), the current SOTA on I3D, by 0.5%. Furthermore,
Table 3: Performance comparisons (AUC@ROC) between the SOTA methods and our method on ShanghaiTech Campus dataset (Liu et al., 2018). The first group is unsupervised methods, and the rest are weakly-supervised methods. Sultani et al. (2018)* is retrained with I3D features as it was previously not evaluated on ShanghaiTech Campus.

<table>
<thead>
<tr>
<th>Sup.</th>
<th>Method</th>
<th>Venue</th>
<th>Feature</th>
<th>AUC@ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un-</td>
<td>Hasan et al. (2016)</td>
<td>CVPR’16</td>
<td>-</td>
<td>60.85</td>
</tr>
<tr>
<td></td>
<td>Gao et al. (2021)</td>
<td>ICCV’19</td>
<td>-</td>
<td>71.20</td>
</tr>
<tr>
<td></td>
<td>Yu et al. (2020)</td>
<td>MM’20</td>
<td>-</td>
<td>74.48</td>
</tr>
<tr>
<td></td>
<td>GCL (Zaheer et al., 2022)</td>
<td>CVPR’21</td>
<td>ResNext</td>
<td>78.93</td>
</tr>
<tr>
<td>Weakly-</td>
<td>GCN (Zhong et al., 2019)</td>
<td>CVPR’19</td>
<td>TSN</td>
<td>84.44</td>
</tr>
<tr>
<td></td>
<td>CLAWS (Zaheer et al., 2020)</td>
<td>CVPR’21</td>
<td>ResNext</td>
<td>86.21</td>
</tr>
<tr>
<td></td>
<td>Purwanto et al. (2021)</td>
<td>ICCV’21</td>
<td>TRN</td>
<td>96.85</td>
</tr>
<tr>
<td></td>
<td>GCN (Zhong et al., 2019)</td>
<td>CVPR’19</td>
<td>-</td>
<td>76.44</td>
</tr>
<tr>
<td></td>
<td>Zhang et al. (2019)</td>
<td>ICIP’19</td>
<td>-</td>
<td>82.50</td>
</tr>
<tr>
<td></td>
<td>CLAWS (Zaheer et al., 2020)</td>
<td>ECCV’20</td>
<td>C3D</td>
<td>89.67</td>
</tr>
<tr>
<td></td>
<td>RTFM (Tian et al., 2021)</td>
<td>ICCV’21</td>
<td>-</td>
<td>91.57</td>
</tr>
<tr>
<td></td>
<td>BN-SVP (Sapkota &amp; Yu, 2022)</td>
<td>CVPR’22</td>
<td>-</td>
<td>96.00</td>
</tr>
<tr>
<td></td>
<td>Sultani et al. (2018)*</td>
<td>CVPR’18</td>
<td>-</td>
<td>85.33</td>
</tr>
<tr>
<td></td>
<td>DAM (Majhi et al., 2021)</td>
<td>AVSS’21</td>
<td>-</td>
<td>88.22</td>
</tr>
<tr>
<td></td>
<td>AR-Net (Wan et al., 2020)</td>
<td>ICME’20</td>
<td>I3D</td>
<td>91.24</td>
</tr>
<tr>
<td></td>
<td>RTFM (Tian et al., 2021)</td>
<td>ICCV’21</td>
<td>-</td>
<td>97.21</td>
</tr>
<tr>
<td></td>
<td>Wu &amp; Liu (2021)</td>
<td>TIP’21</td>
<td>-</td>
<td>97.48</td>
</tr>
<tr>
<td></td>
<td>Ours: CLIP-TSA†</td>
<td></td>
<td></td>
<td>97.19</td>
</tr>
<tr>
<td></td>
<td>Ours: CLIP-TSA‡</td>
<td></td>
<td></td>
<td>98.32</td>
</tr>
<tr>
<td></td>
<td>Ours: CLIP-TSA‡</td>
<td></td>
<td></td>
<td>97.98</td>
</tr>
</tbody>
</table>

Table 4: Performance comparisons (AUC@PR) between the SOTA methods and our method on XD-Violence dataset (Wu et al., 2020). The first group is an unsupervised method, and the other group is weakly-supervised methods. V and A represent visual and audio features, respectively.

<table>
<thead>
<tr>
<th>Sup.</th>
<th>Modality</th>
<th>Method</th>
<th>Venue</th>
<th>Feature</th>
<th>AUC@PR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Un-</td>
<td>–</td>
<td>OCSVM</td>
<td>Schölkopf et al., 1999</td>
<td>NeuIPS’00</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hasan et al., 2016</td>
<td>CVPR’16</td>
<td>–</td>
</tr>
<tr>
<td>Weakly-</td>
<td>Vision &amp; Audio</td>
<td>Wu &amp; Liu (2021)</td>
<td>TIP’21</td>
<td>I3D(V) + VGGish(A)</td>
<td>75.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wu et al. (2020)</td>
<td>ECCV’20</td>
<td>I3D(V) + VGGish(A)</td>
<td>78.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pang et al. (2021)</td>
<td>ICASSP’21</td>
<td>I3D(V) + VGGish(A)</td>
<td>81.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NICE-LSTM Yang et al. (2022)</td>
<td>MM’22</td>
<td>I3D(V) + VGGish(A)</td>
<td>83.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DFL-Pu &amp; Wu (2022)</td>
<td>ICCECE’22</td>
<td>I3D(V) + VGGish(A)</td>
<td>83.54</td>
</tr>
<tr>
<td></td>
<td>Vision</td>
<td>Sultani et al. (2018)</td>
<td>CVPR’18</td>
<td>C3D(V)</td>
<td>73.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTFM (Tian et al., 2021)</td>
<td>ICCV’21</td>
<td>C3D(V)</td>
<td>75.89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ours: CLIP-TSA†</td>
<td></td>
<td>C3D(V)</td>
<td>77.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTFM (Tian et al., 2021)</td>
<td>ICCV’21</td>
<td>I3D(V)</td>
<td>77.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ours: CLIP-TSA‡</td>
<td></td>
<td>I3D(V)</td>
<td>78.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ours: CLIP-TSA‡</td>
<td></td>
<td>CLIP(V)</td>
<td>82.19</td>
</tr>
</tbody>
</table>

CLIP-TSA yields superior performance to that of Wu & Liu (2021), the current best-performing model, by 0.84% with the end-to-end training scheme.

Lastly, Table 4 shows the frame-level AUC@PR results of SOTA models on the XD-Violence dataset, which is the most recently released dataset of the three. From the table, it can be seen that ours outperforms all SOTA models on various visual features as well as some models that leveraged both visual and auditory features. More specifically, it has left a remarkable margin of 1.77%, 0.38%, and 4.38% on C3D, I3D, and CLIP, respectively.

Our hypothesis for relatively small performance improvement on the ShanghaiTech Campus dataset compared to UCF-Crime and XD-Violence is that optimality has already been achieved in the ShanghaiTech Campus dataset as it is already yielding very high, near-100% scores by SOTA models. As a result, we believe that it is much more difficult to pull up its score in comparison to the remaining two datasets, with problems potentially being noise or subjective, frame-level human label errors.
Moreover, comparing Table 5 with Tables 2-4, we observe that the best-performing baseline model for different batch sizes (4, 8, 16, 32, 64), and three different learning rates (0.01, 0.001, 0.0001).

To ensure the reported performance is generalized enough, we run the model five times each. The performance of the model at each $r$ for each dataset is shown in Figure 3. According to the figure, the value of $r$ where the model performs at optimal level differs for each. For example, UCF-Crime yields 87.6% AUC@ROC when $r$ is set to 0.7, ShanghaiTech Campus obtains 98.3% AUC@ROC at $r \in [0.3, 0.7]$, and XD-Violence gets 82.2 AUC@PR at $r = 0.9$.

To understand why the optimal value of $r$ is changing from dataset to dataset, we collect two pieces of information: 1) data distribution of UCF-Crime (1,900 videos with 13 types of anomalies), ShanghaiTech Campus (317,398 videos with 130 anomaly events), and XD-Violence (4,754 videos with 6 types of anomalies) datasets; 2) frame-level anomaly-to-all ratio ($\frac{\text{# of Anomaly}}{\text{# of Anomaly} + \text{# of Normal}}$) of their test sets, which are 0.1819, 0.4247, and 0.4977, respectively. The ShanghaiTech Campus is a big-scale data with a comparatively large number of anomaly events, so we assume that the model is able to be generalized enough, and the values of $r \in [0.3, 0.7]$ are around the anomalous-to-all ratio. However, UCF-Crime and XD-Violence have imbalanced anomaly category distributions as shown in Figure 4. Furthermore, the important features for making the correct decision are not limited to anomalous snippets, but also include some normal snippets as well, especially as part of computation for loss, in which both the magnitudes for top anomalous snippets and normal snippets are factored in. Thus, the best values of $r$ in UCF-Crime and XD-Violence datasets are not aligned around anomalous-to-all ratios.

Effectiveness of TSA mechanism In order to investigate the effectiveness of TSA, we conduct the experiments in two cases with and without TSA on the same vision-language feature on UCF-Crime, ShanghaiTech Campus, and XD-Violence datasets. Both experiments are sharing the same configuration settings with five different seeds, five different batch sizes (4, 8, 16, 32, 64), and three different learning rates (0.01, 0.001, 0.0001).

Table 5 reports the best performance of both CLIP-TSA and baseline model (w/o TSA) on three separate datasets. From the table, it can be observed that CLIP-TSA outperforms the baseline on all three when it is compared to the same dataset. Moreover, comparing Table 5 with Tables 2-4 we observe that the best-performing baseline model for
Table 5: Ablation study of TSA on UCF-Crime, ShanghaiTech Campus, and XD-Violence Datasets, using the corresponding metric for each. The best score is **bolded**, the runner-up is *underlined*, and the improved score after the TSA is *italicized.*

<table>
<thead>
<tr>
<th>Feature</th>
<th>TSA</th>
<th>UCF-Crime (AUC@ROC ↑)</th>
<th>ShanghaiTech (AUC@ROC ↑)</th>
<th>XD-Violence (AUC@PR ↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3D</td>
<td>✗</td>
<td>82.59</td>
<td>96.73</td>
<td>76.84</td>
</tr>
<tr>
<td>C3D</td>
<td>✓</td>
<td>83.94</td>
<td>97.19</td>
<td>77.66</td>
</tr>
<tr>
<td>I3D</td>
<td>✗</td>
<td>83.25</td>
<td>96.39</td>
<td>77.74</td>
</tr>
<tr>
<td>I3D</td>
<td>✓</td>
<td>84.66</td>
<td>97.98</td>
<td>78.19</td>
</tr>
<tr>
<td>ViT</td>
<td>✗</td>
<td>86.29</td>
<td>98.18</td>
<td>80.43</td>
</tr>
<tr>
<td>ViT</td>
<td>✓</td>
<td>87.58</td>
<td>98.32</td>
<td>82.19</td>
</tr>
</tbody>
</table>

Each dataset is shown to yield a higher score than the SOTA models for the respective dataset. That implies the strength and efficiency of vision-language in VAD.

## 5 Conclusion

This paper presents CLIP-TSA, an effective end-to-end weakly-supervised VAD framework. Specifically, we proposed the novel TSA mechanism that maximizes attention on a subset of features while minimizing attention on noise and showed its applicability to the weakly-supervised VAD problem. We also applied TSA to CLIP-extracted features to demonstrate its efficacy in Visual Language features and exploited visual language features in the weakly-supervised VAD problem. We also empirically validate the excellence of our model on the three popular VAD datasets by comparing ours against the SOTAs.

Future investigations might aim for better techniques to incorporate both temporal and spatial information as well as handle imbalanced data with less annotation. Techniques for attention such as (Li et al., 2022) and self-supervised learning (Caron et al., 2021; Chen et al., 2020a) are also potential extensions for performance improvement.
References


