Supplementary material: "Generative Cooperative Learning for Unsupervised Video Anomaly Detection"

M. Zaigham Zaheer[1,2,3,5], Arif Mahmood[4], M. Haris Khan[5], Mattia Segù[3], Fisher Yu[3], Seung-Ik Lee[1,2]
Electronics and Telecommunications Research Institute[1], Univ. of Science and Technology[2], ETH Zurich[3], Information Technology Univ. Punjab[4], Mohamed bin Zayed Univ. of Artificial Intelligence[5]

This is the supplementary material for the paper titled, 'Generative Cooperative Learning for Unsupervised Video Anomaly Detection'. It includes additional analysis of various design choices as well as more qualitative results, accompanying Sec. 4 of the main paper.

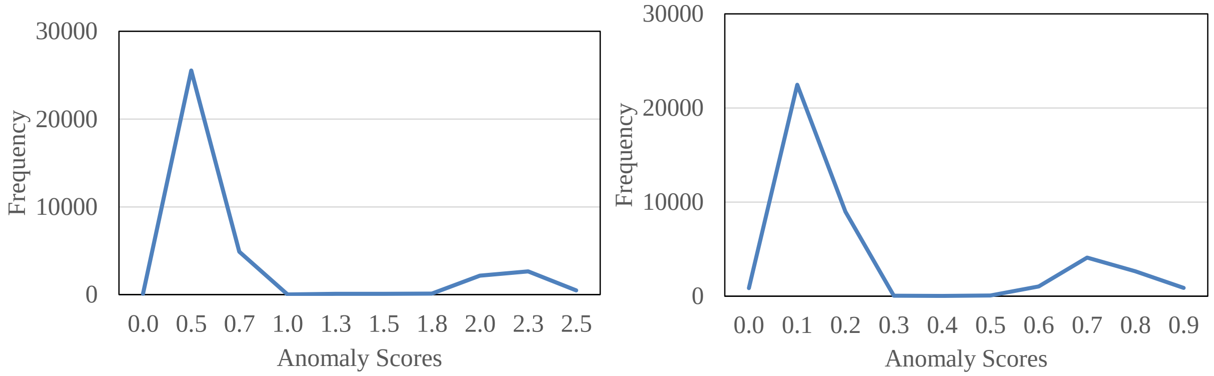
1. Threshold methods

As discussed in Section 3.2 of the main paper, we utilize pseudo-labels created from the discriminator $\mathcal{D}$ to train the Generator $\mathcal{G}$, and vice versa. In order to create pseudo-labels, we apply a pre-defined thresholds $\mathcal{L}_G^{th}$ and $\mathcal{L}_D^{th}$. Intuitively, there are several different possibilities to define these thresholds. In this study, we explore three different ways including *hard-coded* values as well as data distribution dependent values.

In the set of *hard-coded* values, we explore to utilize threshold based on the score predicted by the network (referred as *score threshold* hereafter) or to set aside a fixed percentage of input feature vectors as anomalous (referred as *percentage threshold* hereafter). Precisely, top 20% of the instances in a batch are declared anomalous when generating pseudo-labels from $\mathcal{G}$ and top 5% instances are declared anomalous when generating pseudo-labels from $\mathcal{D}$. Although, our method is not strictly dependent on these hyper-parameters, we have observed that in general keeping the percentage low for anomalous instances in generating pseudo-labels from $\mathcal{D}$ yields a slightly better performance, which may be attributed to the use of negative learning. An analysis on increasing the anomalous examples is also provided in the subsequent discussion. In *score threshold*, a fixed threshold of 0.5 over the scores of both $\mathcal{G}$ and $\mathcal{D}$ is applied. This way, any feature vector scoring higher than this value is considered anomalous. Note that the output scores are normalized over an input batch. In *percentage threshold*, we define a given percentage of feature vectors in the input batch as anomalous. This configuration makes the score distribution irrelevant and top scoring features are automatically selected. A similar approach has been used by Sultani *et al.* [1] in which each video is converted into a bag of fix number of segments and top few segments of a video are considered anomalous. However, in our case, we do not assume any fix number of segments. Moreover, as we do not assume any labels towards training, the threshold is applied on the input batch rather than a given video. There are a few examples in the existing literature which assume that a random batch sampled from a given video may contain anomalies [2,4]. However these approaches eventually use video-level labels to identify the batches in which anomalies might be present. We, on the other hand, do not use any supervision. Therefore, we sample a batch from the whole dataset and utilize batch based statistics to create psuedo-labels.

Finally, we also explore a data driven approach in which anomalies are defined if the anomaly score of a particular instance is higher than $\mu + a(\sigma)$ of the score distribution, where μ is mean of the score distribution, σ is standard deviation of the score distribution, and a is a parameter that controls the overall placement of threshold over the given score distribution. This method is referred as *data driven threshold* hereafter. The intuition behind this approach is that most of the data is usually normal and therefore, the mean of the data would lie near the mean of the normal data. Hence, as we move farther from this mean, more anomalous instances may be separated successfully without including normal instances.

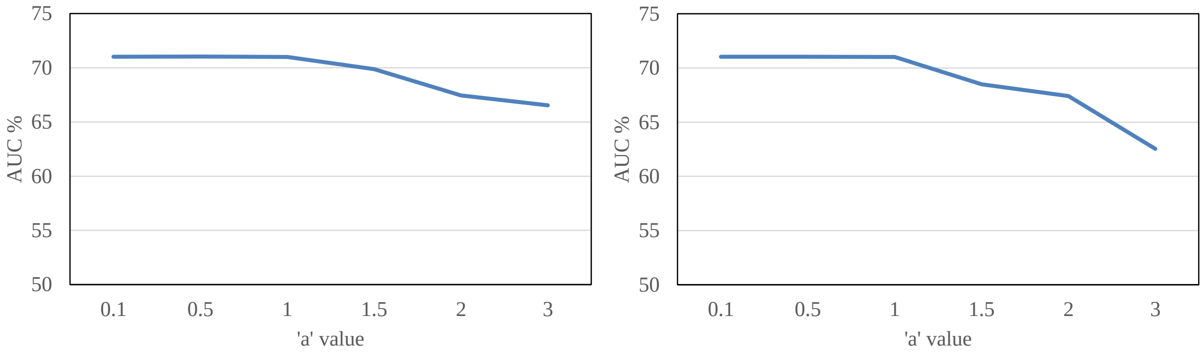
We compare the three threshold methods in Table 1. As seen, *percentage* and *data driven* threshold methods demonstrate an identical performance which is due to the similar effect achieved by both approaches, i.e., dividing high scoring normal instances from the anomalous instances (Fig. 1). In our experiments (Table 1), we found that *percentage* and *data driven* threshold can be used interchangeably without losing any performance. Moreover, score threshold also performs comparably. In order to explore this trend further, in Fig. 1, we also visualize the score distributions of $\mathcal{G}$ and $\mathcal{D}$ in a training batch. Two peaks are clearly visible at both ends of the score distribution. Assuming one of the distribution contains scores from mostly



(a) Scores distribution of an under-trained $\mathcal{G}$

(b) Scores distribution of an under-trained $\mathcal{D}$

Figure 1. Scores distributions of $\mathcal{G}$ and $\mathcal{D}$ being trained in our proposed approach. X-axis represents the anomaly scores and y-axis represents the frequency of these scores happening for the input data. As seen, in both models, two peaks are visible which may presumed to be normal and anomalous. As long as our threshold method dissects the distribution in the middle of the two peaks, the performance of our approach remains comparable.



(a) Effects of threshold variation on $\mathcal{G}$

(b) Effects of threshold variation on $\mathcal{D}$

Figure 2. Effects of threshold variations is studied for $\mathcal{G}$ and $\mathcal{D}$ in our proposed system. As long as the threshold remains in a reasonable range of [0, 1], the performance remains comparable.

normal instances and the other contains scores from mostly anomalous instances, each of the threshold method explored in our study can successfully separate the two distributions, thereby pseudo-labeling one as normal and the other as anomalous.

As the quantity of anomalies in a given system is unknown, it is important that the system is robust to a specific threshold value. We examine our approach for this property by varying the threshold values by controlling the parameter a. A higher a value, e.g. 1, essentially means that the threshold is placed higher ($1 \times \sigma$ away from the mean) towards the anomaly side of the score distribution therefore, less instances are pseudo-labeled as anomalous. Fig. 2 summarizes the results of the study. As seen, the proposed system performs almost identical when a is less than 1. However, the performance drops drastically with a higher value of a. It is because with higher a value, the threshold value is places farther from the mean of the score distribution towards the anomaly side. This results in more anomaly samples pseudo-labeled as normal, which consequently degrades the performance. On the other hand, a lower than 1 value of a does not affect drastically as the data has abundant normal samples and some mislabeled normal samples do not impact on the training as much. In general, as long as the threshold value stays within the two peaks shown in Fig. 1, our proposed approach performs reasonably.

2. Additional Qualitative Results

Here, we present additional qualitative results, accompanying Sec. 4 in the main paper. Fig. 3 and Fig. 4 plot anomaly scores produced by AE$_{AllData}$ and $\mathcal{G}$ & $\mathcal{D}$ trained in our GCL framework on several videos (from UCF-crime dataset). We can notice that, despite being trained in an unsupervised mode, $\mathcal{G}$ trained in our GCL approach often produces better anomaly

Table 1. Comparison of several threshold methods for generating pseudo-labels.

Threshold Method	**AUC %**
Percentage threshold	71.04
Score threshold	70.63
Data driven threshold	71.04

scores compared to $\text{AE}_{AllData}$. Furthermore, since $\mathcal{D}$ is often more robust to noisy data [3, 4], it produces significantly improved results in almost all cases.

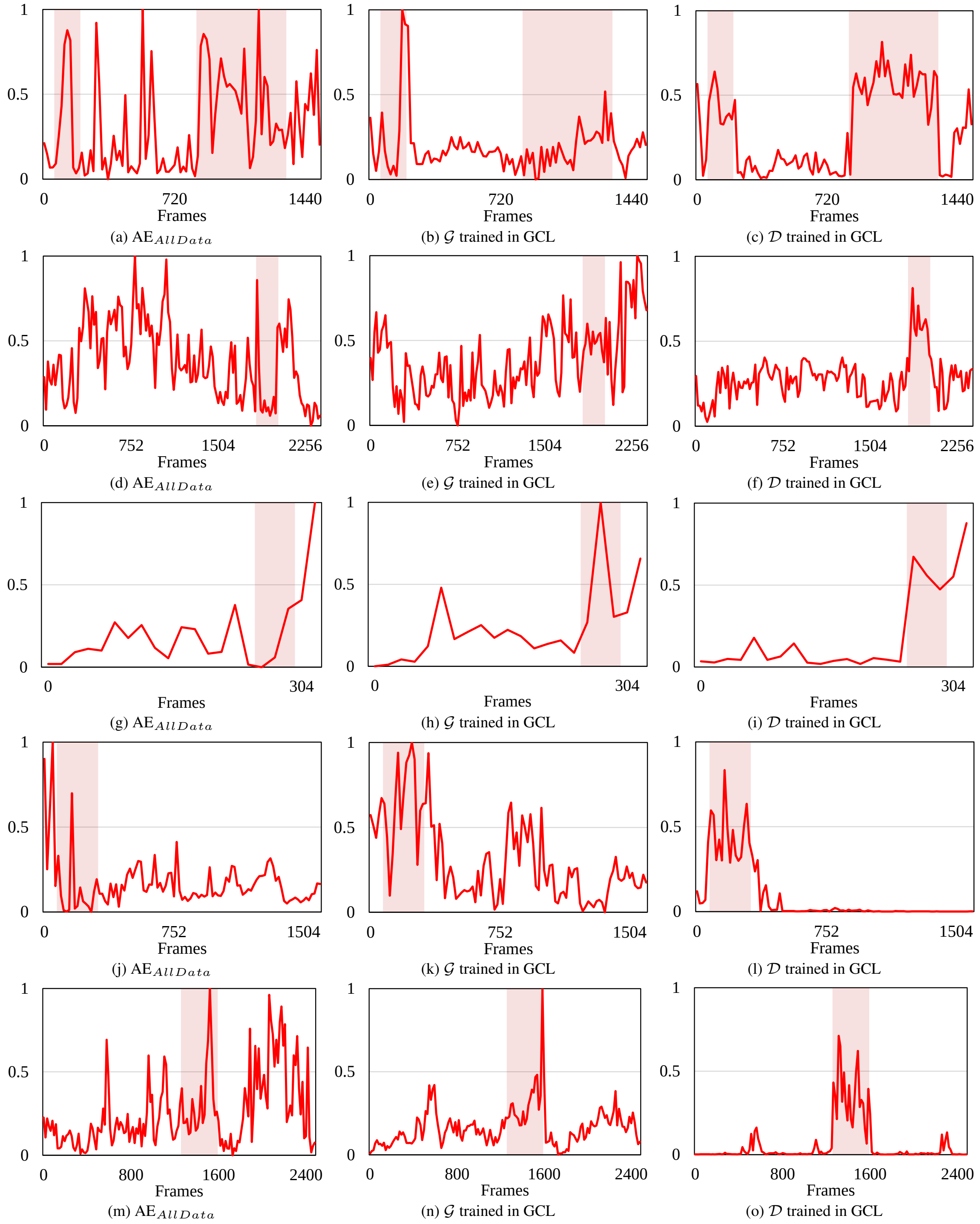
References

[1] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018. 1

[2] Muhammad Zaigham Zaheer, Jin-ha Lee, Marcella Astrid, Arif Mahmood, and Seung-Ik Lee. Cleaning label noise with clusters for minimally supervised anomaly detection. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020. 1

[3] Muhammad Zaigham Zaheer, Jin Ha Lee, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Stabilizing adversarially learned one-class novelty detection using pseudo anomalies, 2022. 3

[4] Muhammad Zaigham Zaheer, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In *European Conference on Computer Vision*, pages 358–376. Springer, 2020. 1, 3



(a) AE$_{AllData}$

(b) $\mathcal{G}$ trained in GCL

(c) $\mathcal{D}$ trained in GCL

(d) AE$_{AllData}$

(e) $\mathcal{G}$ trained in GCL

(f) $\mathcal{D}$ trained in GCL

(g) AE$_{AllData}$

(h) $\mathcal{G}$ trained in GCL

(i) $\mathcal{D}$ trained in GCL

(j) AE$_{AllData}$

(k) $\mathcal{G}$ trained in GCL

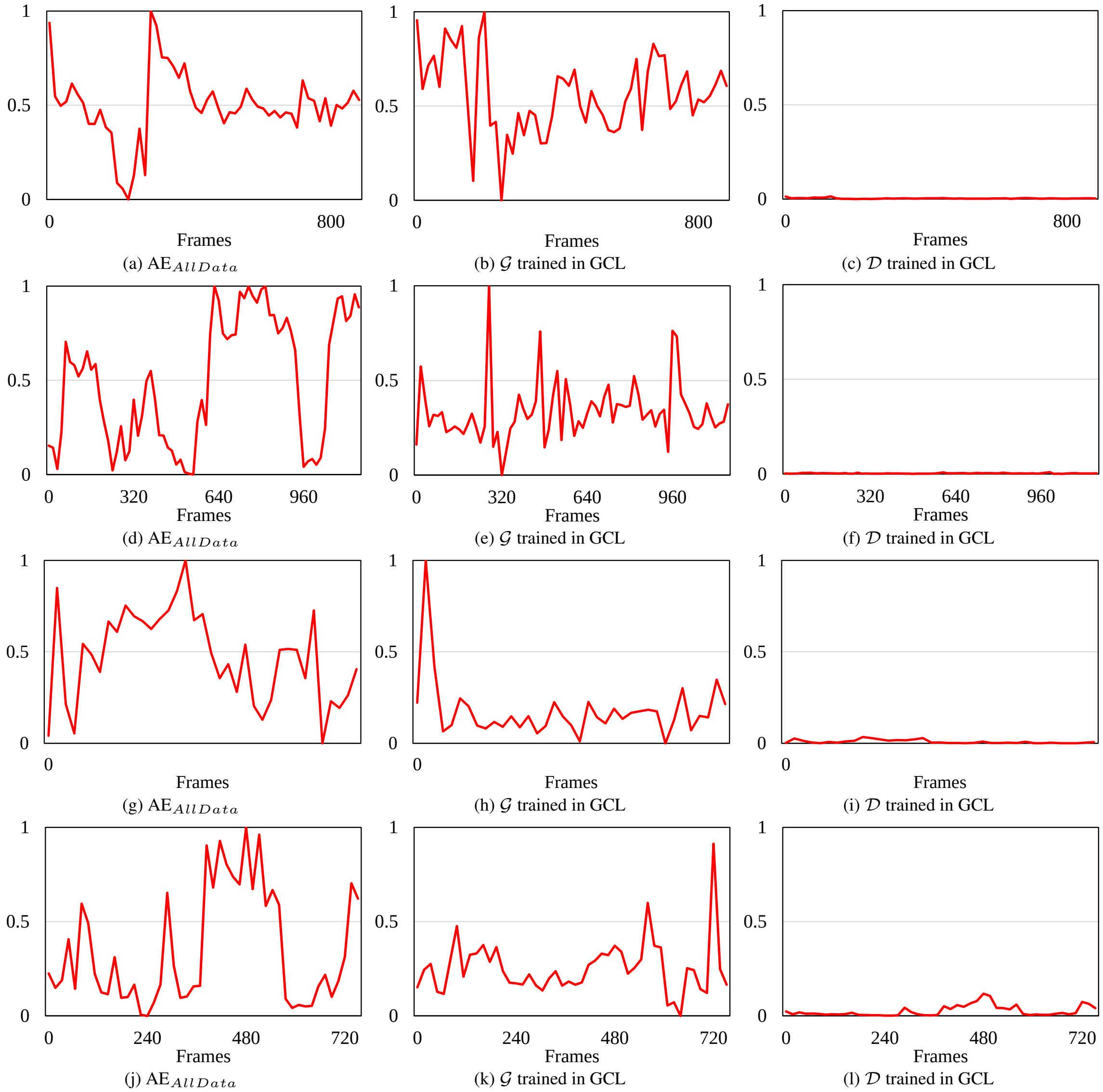
(l) $\mathcal{D}$ trained in GCL

(m) AE$_{AllData}$

(n) $\mathcal{G}$ trained in GCL

(o) $\mathcal{D}$ trained in GCL

Figure 3. Anomaly scores plotted on $burglary021$ ((a), (b), and (c)), $explosion029$ ((d), (e), and (f)), $roadaccident002$ ((g), (h), and (i)), $shooting008$ ((j), (k), and (l)), and $stealing036$ ((m), (n), and (o)) of UCF-Crime by AE$_{AllData}$, $\mathcal{G}$ trained in GCL, and $\mathcal{D}$ trained in GCL, respectively. In all cases cases $\mathcal{G}$ performs more reasonably compared to AE$_{AllData}$, $\mathcal{D}$ successfully produces discriminative scores for most normal and anomalous parts of the videos. In (f), scores for normal portions of the video also stay relatively higher. However, compared to the other two models in (d) and (e), the overall scores are significantly separable.



(a) $\mathrm{AE}_{AllData}$

(b) $\mathcal{G}$ trained in GCL

(c) $\mathcal{D}$ trained in GCL

(d) $\mathrm{AE}_{AllData}$

(e) $\mathcal{G}$ trained in GCL

(f) $\mathcal{D}$ trained in GCL

(g) $\mathrm{AE}_{AllData}$

(h) $\mathcal{G}$ trained in GCL

(i) $\mathcal{D}$ trained in GCL

(j) $\mathrm{AE}_{AllData}$

(k) $\mathcal{G}$ trained in GCL

(l) $\mathcal{D}$ trained in GCL

Figure 4. Anomaly scores plotted on $normal904$ ((a), (b), and (c)), $normal905$ ((d), (e), and (f)), $normal907$ ((g), (h), and (i)), and $normal911$ ((j), (k), and (l)) of UCF-Crime by $\mathrm{AE}_{AllData}$, $\mathcal{G}$ trained in GCL, and $\mathcal{D}$ trained in GCL, respectively. $\mathcal{D}$ produces noticeably low scores smoothly in the normal videos demonstrating superior performance compared to the other two models.