Interleaving One-Class and Weakly-Supervised Models with Adaptive Thresholding for Unsupervised Video Anomaly Detection

Yongwei Nie[1], Hao Huang[1], Chengjiang Long[2], Qing Zhang[3], Pradipta Maji[4], and Hongmin Cai[1]

[1] South China University of Technology, China
[2] Meta Reality Labs, USA
[3] Sun Yat-sen University, China
[4] Indian Statistical Institute, India

A Supplementary Material

A.1 wOCC Training Loss for other Employed OCC Models

We have discussed the wOCC training loss of STG-NF [1] in the paper. In what follows, we discuss the wOCC training loss of other employed OCC models: AE [6] and Jigsaw [5].

AE [6]. The basic building block for AE is the snippet feature X extracted by I3D. For improving AE to wOCC, we define its wOCC training loss by:

$$L_{wocc} = (1 - w_X)\|X - f_{AE}(X)\|_2^2 \tag{1}$$

where w_X is the weight of X and $f_{AE}(X)$ returns the reconstructed X by AE.

Jigsaw [5]. As for Jigsaw, the basic building block is a jigsaw puzzle X. The jigsaw puzzle is processed in two streams: shuffled in spatial and temporal. Spatially, each frame is decomposed into $n \times n$ patches which are then shuffled. All the frames share the same permutation meanwhile are kept in chronological order. Temporally, Jigsaw shuffles the frame sequence containing l frames without disorganizing the spatial content. Its wOCC training loss based on cross-entropy (CE) loss is defined as:

$$L_{wOCC} = \begin{cases} (1 - w_X)\frac{1}{l}\sum_{i=1}^{l} CE(t_i, \hat{t_i}), & X \in Q_t \\ (1 - w_X)\frac{1}{n^2}\sum_{j=1}^{n^2} CE(s_j, \hat{s_j}), & X \in Q_s \end{cases}, \tag{2}$$

where w_X is the weight of X, and Q_s and Q_t respectively denote the sets of spatial and temporal jigsaw puzzles. What's more, t_i and $\hat{t_i}$ are the ground-truth and predicted positions of a frame in the original sequence respectively, and s_j and $\hat{s_j}$ are the ground-truth and predicted locations of a patch in the original spatial splitting grid, respectively.

A.2 More Implementation Details

When incorporating different wOCC or WS models into our framework, they are implemented slightly differently for fair comparison or better performance.

AE [6]. All other wOCC and WS models are optimized by Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, except AE. That is because in the work of [6], AE is optimized using RMSprop optimizer with a learning rate of 0.00002 and momentum of 0.60. To compare with [6] fairly, we follow this setting. The architecture of AE is FC[2048,1024,512,256,512,1024,2048]. Due to the simple network, the batch size of the AE is 8192, and we train the model for 15 epochs before swapping to train the WS model.

Jigsaw [5]. For Jigsaw, the learning rate is $1e$-4 and the batch size is 192, following the setting in [5]. We train the network for 5 epochs in each wOCC and WS swapping step.

Sultani et al. [2]. We train the model with a batch size of 32 and the learning rate of $5e$-5, following the setting in [2]. We train the model for one epoch in each alternate training step.

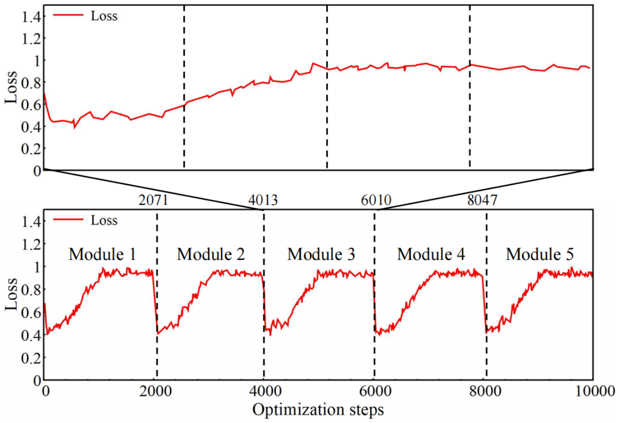

Fig. S1: Bottom: training loss curve of the WS model during the whole repeating procedure. Top: zoomed-in training loss curve in the third module.

A.3 Training Loss Curve of the Weakly-Supervised Model

In Figure S1, we show the training loss curve of the WS (RTFM) model in our framework. Originally in RTFM [3], the loss increases as the training proceeds, in contrast to the usual case where loss is minimized. As seen, our loss in each interleaving training module follows this trend. Note that in each module, we have multiple wOCC and WS alternate training steps. Please check the zoomed-in curve showing the loss curve of the WS model in the third module. At the time of swapping back to train the WS model (indicated by the vertical dash lines in the top figure), the loss does not change abruptly. This shows that though involved into the alternate training, the WS model can converge smoothly (similar to the wOCC model), not affected much by training abrupt swapping of the training.

We also observe that the loss curves in all the modules look similar to each other, and there is no increase in the peak value of the loss magnitude when more modules are performed. This goes against the usual expectation that there should be better (here higher) losses in later training modules. This is because 1 is the maximum loss the adopted WS model can achieve (according to the design of the loss function in [4]). The WS models are trained to their best modes in all the modules.

Table S1: The number of training steps in each interleaving training module with different methods on different datasets.

wOCC Model	WS Model	Dataset	Training Steps				
			Module 1	Module 2	Module 3	Module 4	Module 5
STG-NF	RTFM	ShanghaiTech	3	2	4	4	4
STG-NF	RTFM	UBnormal	5	4	5	5	5
STG-NF	Sultani et al. [2]	ShanghaiTech	2	2	3	3	3
AE	RTFM	ShanghaiTech	2	3	4	3	3
Jigsaw	RTFM	ShanghaiTech	3	4	3	3	3

A.4 The Number of Training Steps in Each Module

As shown in Table S1, we list the number of training steps in each interleaving training module with different methods on different datasets in detail. The number of training steps in each module depends on the module convergence conditions we set. Once the change of the training loss between each training step is lower than $U = 0.1$, interleaving training stops. Please check the first and the second rows in Table S1. With the same wOCC and WS models, our method shows different convergence speeds on different datasets due to the different compositions of the datasets. What is more, we also present the number of training steps in each module on ShanghaiTech after replacing the wOCC or WS model (See the third to fifth rows in Table S1). It is natural to have different convergence speeds with different models.

There are some common trends for the number of training steps with different methods on different datasets. Please check the columns "Module 4" and "Module 5" in Table S1. Take the first row as an example. The number of training steps in the fourth and fifth module are the same. The same phenomenon exists in other rows as well. This indicates that as the training process progresses, the convergence rate of our method can gradually stabilize within each module, indicating the stability of our method.

A.5 More Convergence Analysis of Threshold T_{ws}

In Figure S2, we conduct more experiments to demonstrate the convergence of the threshold T_{ws} given different $R\%$. At the top, the dataset is UBnormal, the

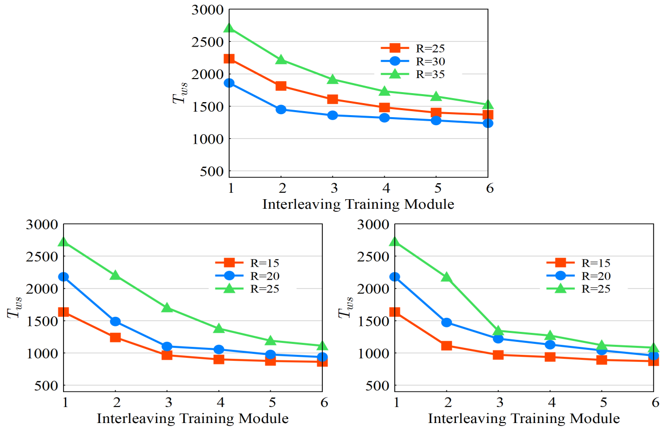

Fig. S2: Top: Using STG-NF as the wOCC model and RTFM as the WS model, different $R\%$ converge to similar T_{ws} on UBnormal. Bottom left: Using Jigsaw as the wOCC model and RTFM as the WS model, different $R\%$ converge to similar T_{ws} on ShanghaiTech. Bottom right: Using STG-NF as the wOCC model and Sultani et al. [2] as the WS model, different $R\%$ converge to similar T_{ws} on ShanghaiTech.

Table S2: The comparison between our method and interleaving two homogeneous models on ShanghaiTech. The framework starts from training Model 1 and Model 1 provides pseudo labels for supervising the training of Model 2. Then, trained Model 2 inversely generates pseudo annotations for Model 1, forming the interleaving training. First row: our method. Second row: interleaving two OCC models. Third row: interleaving two WS models.

Interleaving Model Types	Model 1	Model 2	Model 1 AUC%	Model 2 AUC%
OCC and WS	STG-NF	RTFM	82.57	88.18
two OCC	STG-NF	AE	80.45	61.23
two WS	RTFM	Sultani et al. [2]	67.23	57.26

wOCC model is STG-NF, and the WS model is RTFM. At the bottom-left, the dataset is ShanghaiTech, the wOCC model is Jigsaw, and the WS model is RTFM. At the bottom-right: the dataset is ShanghaiTech, the wOCC model is STG-NF, and the WS model is [2]. As can be seen, despite variations in convergence speed, different initializations of $R\%$ converge to similar T_{ws}. These experiments also demonstrate that the convergence of T_{ws} is not dependent on specific models or datasets.

A.6 Interleaving Training Two Homogeneous Models

Typically, a UVAD method is implemented by training two VAD models. We choose to interleave OCC and WS methods, but there are potential options to interleave two homogeneous models, e.g., two OCC models. How about interleaving two homogeneous models for tackling UVAD?

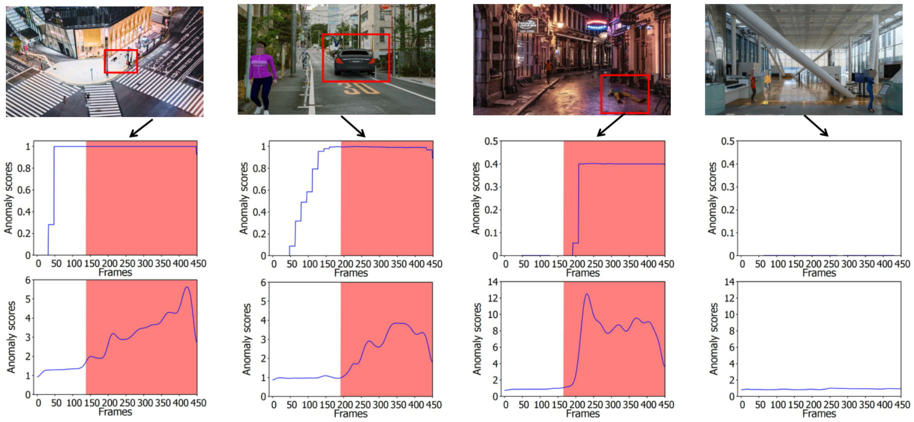

Fig. S3: Qualitative results on UBnormal. Left to right column: visual results of the videos *abnormal*_*scene*_12_*scenario*_8, *abnormal*_*scene*_29_*scenario*_4, *abnormal*_*scene*_3_*scenario*_6 and *normal*_*scene*_5_*scenario*_8. Middle row: anomaly score curves performed by the WS (RTFM) model. Bottom row: anomaly score curves performed by the wOCC (STG-NF) model. Red square area: the interval where anomalies occur in the video. Blue curve: the computed anomaly score of video frames.

We first discuss combining two OCC models (STG-NF and AE) into a unified training framework. The training framework starts from STG-NF, since compared with AE, STG-NF is more robust and has stronger performance. As shown in Table S2, the best AUC is 80.45% in such framework which is lower than 88.18% in our method. We conjecture that interleaving two OCC methods just models normal data but overlooks valuable anomalies, causing lower performance. The situation is worse when interleaving two WS models (RTFM and Sultani et al. [2]) compared with interleaving two wOCC models. The best AUC of the framework is only 67.23% and what's more, the training of the framework cannot converge on the UVAD dataset, yielding the AUC of 64.12% on RTFM after repeating the interleaving modules 5 times. WS models require relatively reliable labels to supervise their training, while at the beginning of interleaving two WS models, there is no reliable way to meet such requirements.

However, the aforementioned problems in interleaving two homogeneous models do not exist in our method thanks to the anomaly modeling of the WS model and relatively reliable labels provided by the wOCC model at the start of training our framework. In conclusion, interleaving two homogeneous models is less effective compared with our method.

A.7 More Visual Results

We further provide more visual results in Figure S3 and Figure S4. In both figures, the middle row is the anomaly score curves performed by the WS (RTFM)

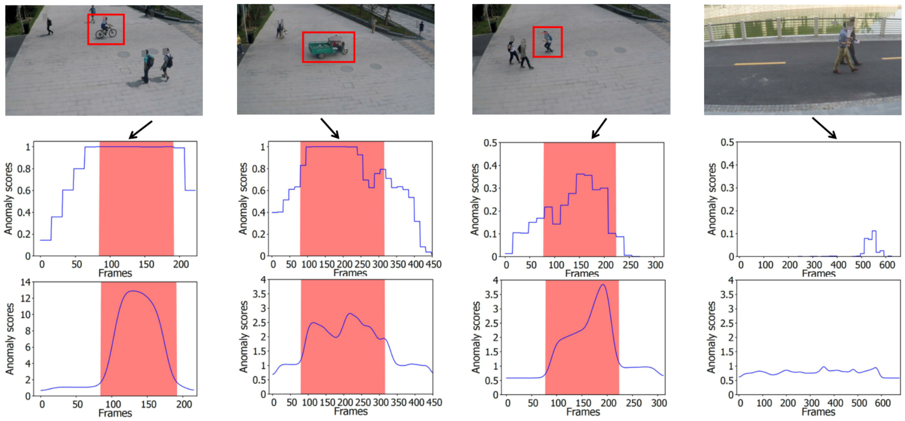

Fig. S4: Qualitative results on ShanghaiTech. Left to right column: visual results of the videos 01_0139, 01_0053, 01_0141 and 11_003. Middle row: anomaly score curves performed by the WS (RTFM) model. Bottom row: anomaly score curves performed by the wOCC (STG-NF) model. Red square area: the interval where anomalies occur in the video. Blue curve: the computed anomaly score of video frames.

model and the bottom row is the anomaly score curves performed by the wOCC (STG-NF) model.

As seen in Figure S3, we show visual results on UBnormal. No matter anomalies of a man running on the road (*abnormal*_*scene*_12_*scenario*_8), a car knocking against the pedestrian (*abnormal*_*scene*_29_*scenario*_4) or a man suddenly falling and convulsing on the ground (*abnormal*_*scene*_3_*scenario*_6), both wOCC and WS models reach a consensus and detect the anomaly intervals correctly. What is more, for the normal event of two women talking with each other (*normal*_*scene*_5_*scenario*_8), both two models do not mistake normal events for abnormal events.

We also show visual results on ShanghaitTech in Figure S4. For the anomaly occurring on the sidewalk of a man riding (01_0139), a tricycle driving (01_0053) and a woman skating on a skateboard (01_0141), wOCC and WS models in our framework predict relatively accurate anomaly scores of high in abnormal intervals and low in normal intervals. Meanwhile, both two models predict low anomaly scores for the normal event of two men walking on the footwalk (11_003).

A.8 Failure Cases

Figure S5 and Figure S6 show some failure cases. For both figures, the middle row is the anomaly score curves performed by the WS (RTFM) model and the bottom row is the anomaly score curves performed by the wOCC (STG-NF) model, too.

We show some failure cases on UBnormal in Figure S5. Sometimes, both wOCC and WS models neglect the same anomaly events. In the left picture, when a man runs across the corridor, both the wOCC and WS models neglect the anomaly due to the man's body being partially obstructed by the desk and being too far from the camera. In contrast to this, both two models easily detect the anomaly of a man twitching on the ground in the same video which is not obstructed by anything and is close to the camera. We conjecture that both occlusion and distance can reduce the extraction and tracking accuracy of human pose features, leading to the failure of our anomaly detection method based on these features. However, there are also cases where one model identifies anomalies while the other does not. In the right picture, the wOCC model accurately detects the anomaly of a man running at a slow speed from right to left at a zebra crossing while the WS model fails. Later, when another man runs from left to right at a zebra crossing at a fast pace, both models agree that this is an anomaly event again. We speculate that the failure of the WS model is attributed to its limited capability to distinguish between running with smaller movements and walking. While for the wOCC model, it can overcome this issue because it conducts a detailed analysis and tracking of body postures.

In Figure S6, we show some failure cases on ShanghaiTech. Similarly, we present an anomaly case where both wOCC and WS models miss and another case where one succeeded in judgment while the other failed. In the left of Figure S6, both models successfully detect the anomalous event where a woman rides a bicycle on the sidewalk when it shows in the center of the frame. However, when she rides away from the camera, both models fail to find out it is still abnormal. We suppose that anomalies occurring further away from the camera exhibit smaller movement amplitudes and are more challenging to track human poses. This results in suboptimal performance for both models. In the right picture, when a car drives across the footway, the WS model detects it as an abnormal event easily while the wOCC model does not. This is caused by the limitation of STG-NF. STG-NF detects anomalies based on tracked human poses, so it can not detect anomalies not related to humans.

In conclusion, our approach is limited by the wOCC and WS models we use. How to better integrate the inference results of both models to complement each other's shortcomings will be investigated in our future work.

References

1. Hirschorn, O., Avidan, S.: Normalizing flows for human pose anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13545–13554 (2023)
2. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6479–6488 (2018)
3. Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J.W., Carneiro, G.: Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4975–4986 (2021)

4. Wan, B., Fang, Y., Xia, X., Mei, J.: Weakly supervised video anomaly detection via center-guided discriminative learning. In: 2020 IEEE international conference on multimedia and expo (ICME). pp. 1–6. IEEE (2020)
5. Wang, G., Wang, Y., Qin, J., Zhang, D., Bao, X., Huang, D.: Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles. In: European Conference on Computer Vision. pp. 494–511. Springer (2022)
6. Zaheer, M.Z., Mahmood, A., Khan, M.H., Segu, M., Yu, F., Lee, S.I.: Generative cooperative learning for unsupervised video anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14744–14754 (2022)

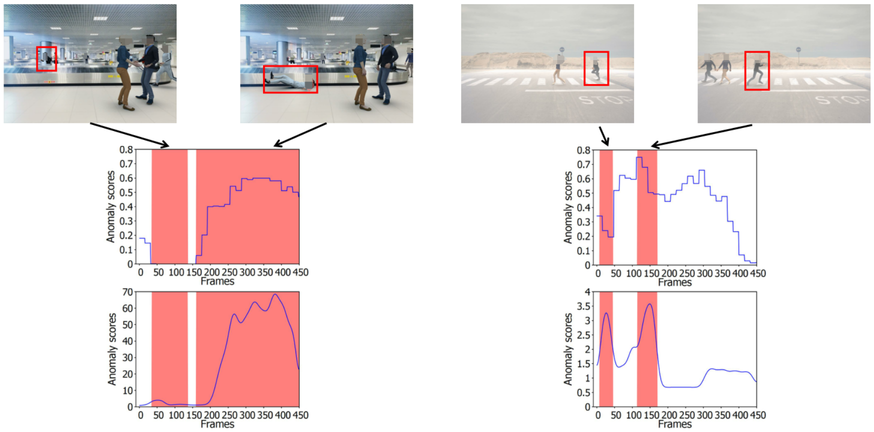

Fig. S5: Failure cases of our framework on UBnormal. Middle row: anomaly score curves performed by the WS (RTFM) model. Bottom row: anomaly score curves performed by the wOCC (STG-NF) model. Left: Both the wOCC and WS models find the anomaly of a man twitching on the ground but fail to detect the anomaly of a running man far from the camera in the video *abnormal_scene_21_scenario_1*. Right: The wOCC model detects both anomaly intervals while the WS model neglects the first man who runs at a smaller pace in the video *abnormal_scene_11_scenario_2_fog*. Red square area: the interval where anomalies occur in the video. Blue curve: the computed anomaly scores of video frames.

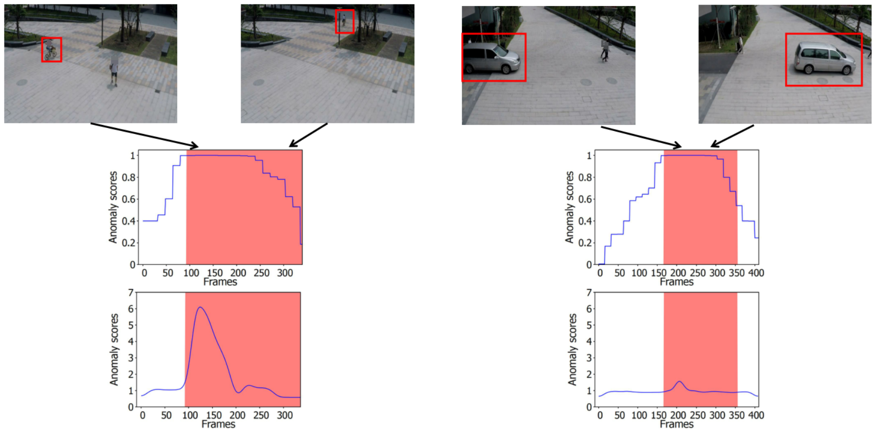

Fig. S6: Failure cases of our framework on ShanghaiTech. Middle row: anomaly score curves performed by the WS (RTFM) model. Bottom row: anomaly score curves performed by the wOCC (STG-NF) model. Left: Both the wOCC and WS models notice the anomaly of a woman riding on the sidewalk in the center of the frame but fail to track the anomaly when the woman riding far from the camera progressively in the video 12_0174. Right: When a car driving on the sidewalk in the video 01_035, the WS model detects the anomaly easily while the wOCC model fails to detect it owing to the limitation of the model. Red square area: the interval where anomalies occur in the video. Blue curve: the computed anomaly scores of video frames.