Supplementary Materials: Video Anomaly Detection with Motion and Appearance Guided Patch Diffusion Model

Anonymous submission

In this supplementary material, we first provide details about our method. We then present additional experimental results to validate our MA-PDM model. Finally, we display more qualitative detection results.

Table 1: The network architecture's hyperparameters.

Setting	Parameter
Batch Size	16
Patch Size	64
Base Channels	64
Channel multipliers	[1,2,2,4]
Attention resolution	16
ResNet-Block layer	1
Encoder channel	64
Encoder chan. mul.	[1,2,2,4]
Encoder channel	64
Encoder chan. mul.	[1,2,2,4]
Decoder chan. mul.	[4,2,2,1]
Memory size	$16 \times 64 \times 256$
β scheduler	Linear
Training T	1000
Diffusion loss	MSE with noise prediction ϵ

1 Details of MA-PDM

Appearance Encoder Network. Table 1 provides the detailed hyperparameters for our MA-PDM model. The Encoder module shares the same layers and channels as the Diffusion U-Net Encoder. Additionally, our patch-memory banks, which are designed to learn appearance normality, are depicted in Figure 1. The output h is derived from the appearance encoder network for a patch image located at coordinates (133,143). We can also select the 11-th memory units by the coordinates for addressing. With the cross-attention operation, we obtain the filtering semantic feature.

Noise Estimate Network. As illustrated in Table 1, the noise estimation network in our approach is lighter compared to the original DDIM (Nichol and Dhariwal 2021). The anomaly detection datasets differ from the ImageNet dataset, having less information. Consequently, we employ smaller channels for learning the noise estimate network. In particular, in the appearance encoder, we refrain from using the self-attention network to mitigate the impact of global

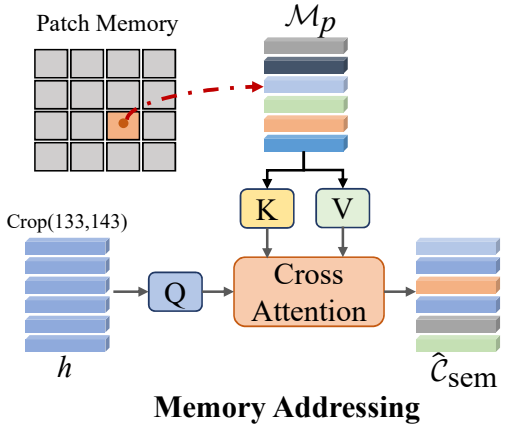


Figure 1: The memory addressing stage of Appearance Encoder.

information on reconstruction. The details of condition fusion are shown in Figure 2, the hierarchical feature maps $[h_1, \ldots, h_n]$ are integrated into the U-Net network. The filtering semantic feature are combined with time embedding and sent to the ResNet-Block of UNet.

2 More Detailed Results

Our model is implemented using PyTorch on a server equipped with a Tesla A100 (40G) GPU. In addition, we provide supplementary experimental results.

Effectiveness of Appearance Network. We conducted experiments across four distinct scenarios aimed at evaluating different approaches to appearance feature embedding on the Ped2 dataset. The summarized outcomes are presented in Table 2. Our MA-PDM only adopts the encoder approach, making use of the encoder characteristic represented as h_e. For reference, we present the AutoEncoder and juxtapose it with our MA-PDM. In this context, h_e and h_d respectively denote the feature maps of the encoder and decoder. Significantly, the results unequivocally underline that the most superior performances are achieved exclusively when employing the encoder design. The AutoEncoder Network is configured with an implementation grounded in the Mean Squared Error (MSE) loss function, with the objective

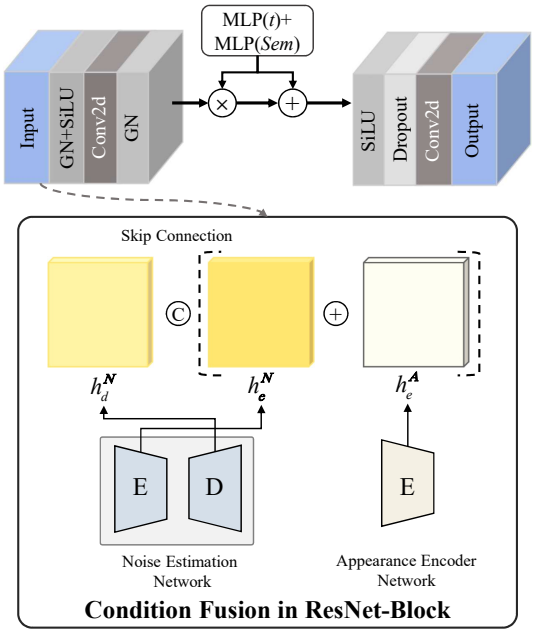


Figure 2: The integration approach of visual characteristic and semantic embedding. E represents the encoder, while D signifies the decoder. The upper section illustrates the visual embedding combined with time embedding. The lower section depicts the fusion process, which dynamically merges the visual feature map with the noise feature map.

of reconstructing the initial patch frames designated as $\mathcal{F}_a$. However, it should be noted that this integration can produce suboptimal reconstruction outcomes, especially in scenarios involving normal patterns. This stands in contrast to the enhanced reconstruction achieved by solely utilizing the Encoder approach.

Table 2: The results of various Appearance networks on the three datasets are outlined in comparison.

Dataset	AutoEncoder		
	h_e	h_d	$[h_e, h_d]$
Ped2	98.6	97.2	98.6
Avenue	91.3	88.2	90.7
Shanghai	79.2	77.2	79.0

FPS. We conducted fps on our method in testing Avenue and Shanghai with batch size of 16, which employs an iterative generation approach, resulting in the current fps rate of 45fps(A100-40G) and 40fps(4090-24G). It's noteworthy that the iterative nature of diffusion methods does indeed pose a notable fps limitation. Nevertheless, we anticipate that as the sampling method continues to undergo refinement and updates, the fps of our method will subsequently increase.

Slide Step. We conducted the results of the different slide steps for the inference stage in Table 3. Small step led to more patches for MA-DPM, but the cost may increase at the same time. We chose Step 64 for our model.

Table 3: Comparison of the results of slide step with the batch size of 16.

Slide-Step	Ped2	Avenue	FPS	GPU
64	98.6	91.3	45	8G
48	98.5	91.3	30	12G
32	98.6	91.2	15	21G

Table 4: Comparison of the results on UCF and XD datasets.

Dataset	MNAD	LGN-Net	Ours
UCF	60.4	64.1	67.3
XD	55.9	57.1	60.4

Complex datasets. As shown in Table 4, we conducted the results on two large and complex datasets: UCF-Crime (Sultani, Chen, and Shah 2018) and XD-Violence (Wu et al. 2020). Unlike the previous weakly supervised methods, we have readjusted the datasets and depart their test sets. We selected 90 normal videos with fewer than 1000 frames from the UCF test sets as training data, 73 abnormal videos with fewer than 1000 frames as test data. In similarity, we selected 83 normal videos with fewer than 2000 frames from the XD test sets as training data, and 132 abnormal videos with fewer than 1000 frames from as test data. The results show our method is capable of handling complex anomaly events in the real world.

3 More Qualitative Results

Anomaly Results. To provide a more comprehensive visualization of the detection results of our method, we performed a detailed comparison of four distinct scenarios, involving our three methodologies, as illustrated in Figure. 3. Importantly, it should be noted that both MNAD (Stephen and Menon 2020) and LGN-Net (Zhao et al. 2022) share a memory-based frame prediction approach similar to our own. The results of our comparison are shown in Figure. 3. The upper part of the figure displays key frames from the videos, with green rectangles indicating normal frames and red rectangles indicating abnormal frames. The bottom part of the figure shows the application of our three modules through four distinct methods. It is clear that our method accurately identifies and highlights the temporal windows that contain anomalies, demonstrating its effectiveness in anomaly detection.

DDIM Steps. The results of our 5-step DDIM time step are shown in Figure 4, which were derived from random noise. The first two lines demonstrate the outcomes of the anomaly prediction, and it is evident that abnormal predictions can be obtained as well. In both typical and atypical scenarios, it is relatively straightforward to replicate ordinary components.

Optical-Flow. We can compare the motion methods of

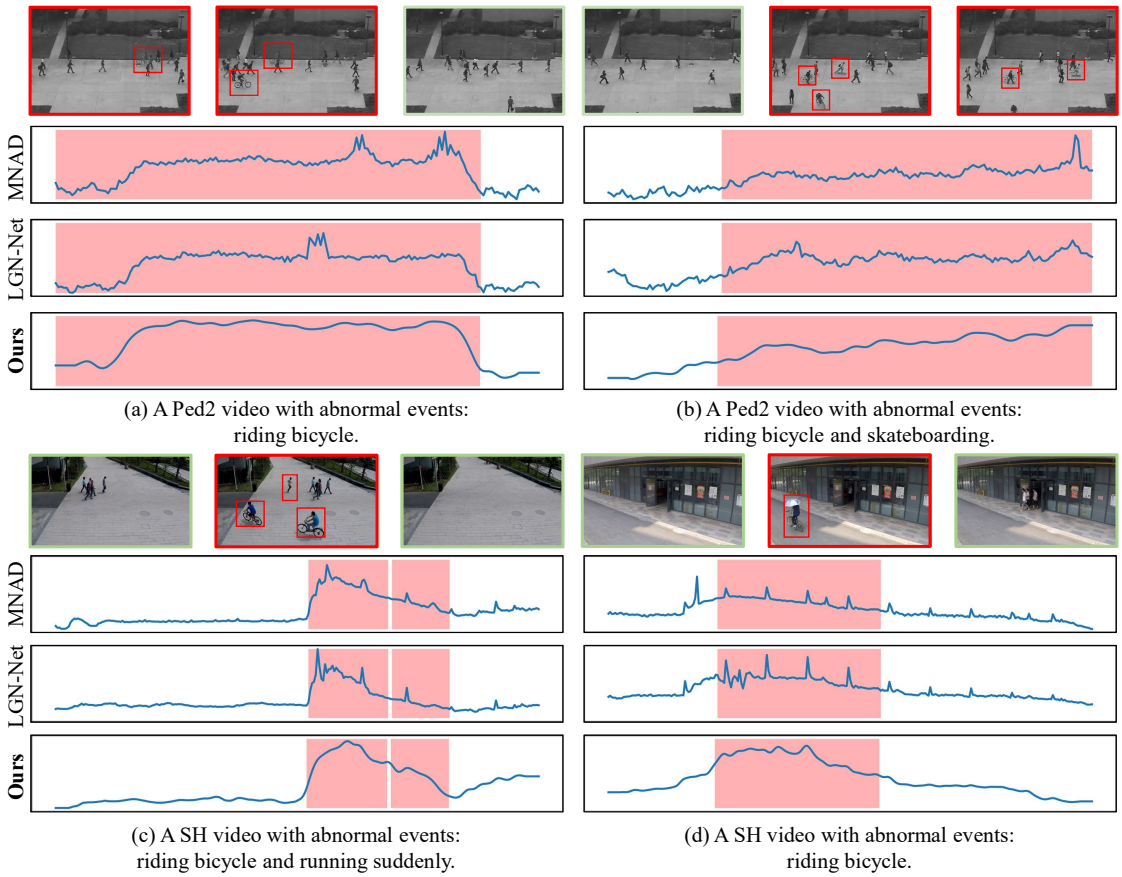


Figure 3: Four examples of anomaly detection comparison on Ped2 and Avenue datasets. From top to bottom, we show the sampled video frames and the results of MNAD, LGN-Net, and Ours. The red regions are abnormal, and the blue curves are the anomaly prediction scores.

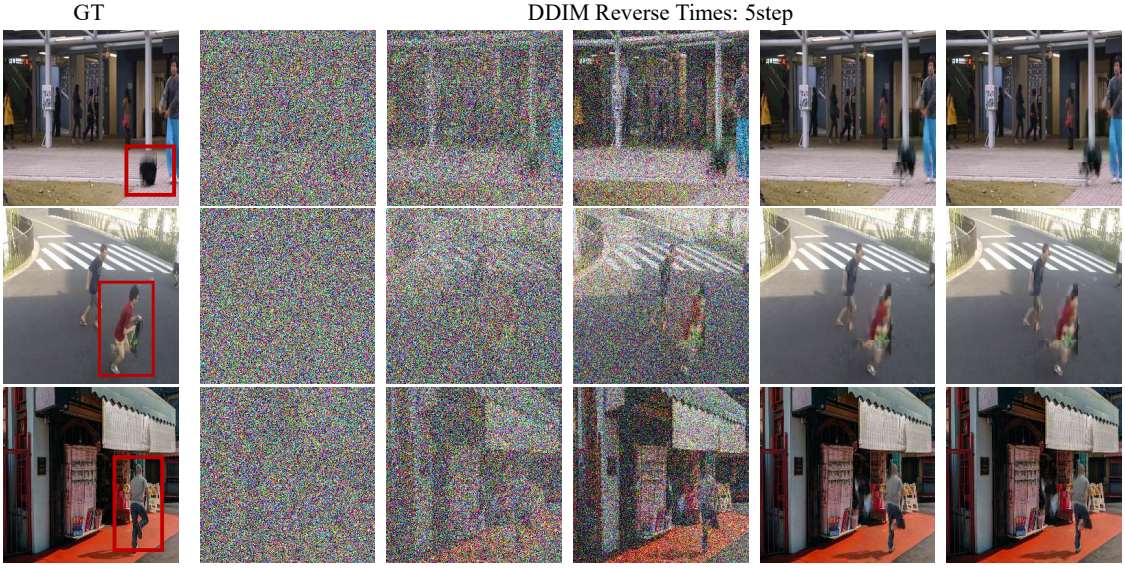


Figure 4: The reverse results of our MA-PDM method in 5-step.

temporal difference (TD) and optical flow (OF) (Teed and Deng 2020) by visualizing the results in Fig. 5. It is evident that the TD motion style is more refined than the OF's.

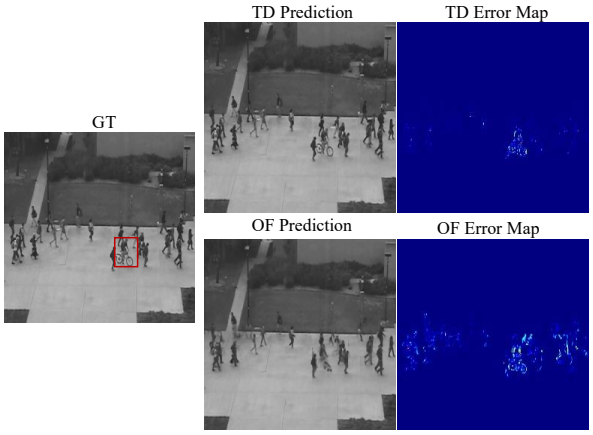


Figure 5: The reverse results of our MA-PDM method in 5-step. The first two lines are abnormal and the last line is normal.

Error Map of predictions. To offer a more insightful elucidation of these results, we present the restructured features of our method in Fig. 6. This visual representation highlights our ability to efficiently reconstruct objects, even when they are in rapid motion, thus serving as evidence of our model's proficiency in effectively addressing swiftly moving objects.

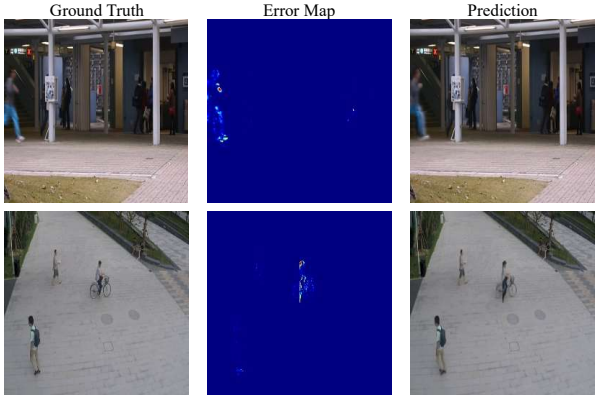
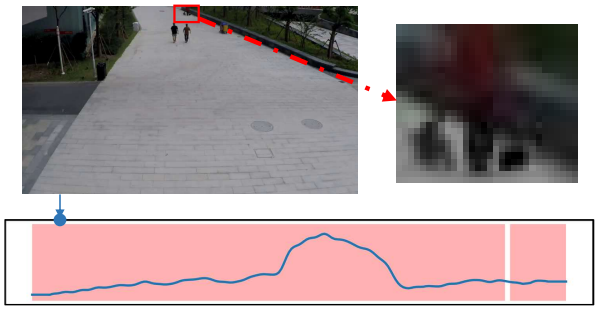


Figure 6: The reverse results of our MA-PDM method in 5-step. The first two lines are abnormal and the last line is normal.

Failed cases. We also showcase instances where our method faces challenges, as illustrated in Fig. 7. These cases shed light on specific limitations within our approach. For example, our model struggles to adequately handle small objects, as evidenced by the instances of cleaning crews in the distance and a bicycle approaching the camera.



(a) A failed case of SH dataset 01_0133

(b) A failed case of SH dataset 06_0145

Figure 7: The reverse results of our MA-PDM method in 5-step. The first two lines are abnormal and the last line is normal.

References

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Stephen, K.; and Menon, V. 2020. Learning Memory-guided Normality for Anomaly Detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 14360–14369.

Sultani, W.; Chen, C.; and Shah, M. 2018. Real-World Anomaly Detection in Surveillance Videos. In *CVPR*, 6479–6488.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 402–419. Springer.

Wu, P.; Liu, J.; Shi, Y.; Shao, F.; Wu, Z.; and Yang, Z. 2020. Not only Look, But Also Listen: Learning Multimodal Violence Detection Under Weak Supervision. In *European Conference on Computer Vision*, 322–339.

Zhao, M.; Liu, Y.; Liu, J.; Li, D.; and Zeng, X. 2022. LGN-Net: Local-Global Normality Network for Video Anomaly Detection. *arXiv preprint arXiv:2211.07454*.