Autoregressive Denoising Score Matching is a Good Video Anomaly Detector Supplementary

Hanwen Zhang*, Congqi Cao*†, Qinyi Lv, Lingtong Min, Yanning Zhang
Northwestern Polytechnical University, Xi'an Shaanxi, 710129, China
zhwwww@mail.nwpu.edu.cn, congqi.cao@mail.nwpu.edu.cn, {lvqinyi, minlingtong, ynzhang}@nwpu.edu.cn

This supplementary material extends the results presented in the main manuscript with additional visualizations and results.

1. Additional Visualizations

To enhance the qualitative analysis of the impact of our approach, we extend the score curves on the NWPU Campus dataset outlined in Section 4.5 of the main manuscript by incorporating the score curves of the original denoising score matching (DSM) method. This addition allows for a more comprehensive comparative evaluation, shedding light on the differences between our method (ADSM), which combines autoregressive denoising score matching with the characteristics of video anomalies, and the original denoising score matching method. This detailed comparison will provide valuable insights into the effectiveness and advantages of our proposed approach in handling the challenging and complex VAD task.

The NWPU Campus [1] is the largest and most complex semi-supervised video anomaly detection benchmark to date. It makes up for the lack of scene-dependent anomalies [12, 13] in the current research field. In the video of "D013_03" within the NWPU Campus dataset, the cyclist is a scene-dependent anomaly, which is normal on the road while abnormal on the square, greatly increasing the difficulty of video anomaly detection. As shown in Fig. 1, the score curve of the DSM dramatically fluctuates wherever the cyclist appears. This indicates that simply using the denoising score matching strategy is not sufficient to detect such anomalies. In contrast, we propose to embed the scene information of the input video sequence into our noise-conditioned score transformer (NCST) to jointly estimate a scene-dependent score, modeling the relationships between video events and scenes. It can be seen in Fig. 1 that the score curve rapidly increases when the cyclist appears on the square while remaining stable at a lower value when the cyclist appears on the road. This demonstrates that our final autoregressive denoising score matching is able to make a timely scene-dependent detection without false alarms, greatly enhancing the adaptability and usability of the video anomaly detection task in smart video surveillance systems.

In the video of "D001_03" within the NWPU Campus dataset, the person climbing the fence is an obvious motion anomaly. As shown in Fig. 1, the score curve of the DSM exhibits a sharp increase upon the occurrence of the fence climber. Nevertheless, it is observed that the score curve fails to sustain a consistently high value after the occurrence, exhibiting frequent fluctuations. This instability raises concerns regarding the likelihood of missed detections or false alarms, highlighting the need for a more robust and reliable detection mechanism to effectively address this issue. In contrast, we propose to assign motion weights to the score function based on the difference between the first and last key frames of the input sequence, guiding our method to focus on the motion consistency inherent in videos. Consequently, the score curve generated by our ADSM exhibits a rapid and sustained increase in value as soon as the person begins climbing the fence, maintaining stability throughout the duration of the abnormal event. This consistent behavior underscores the superior capability of our method in capturing and comprehending the motion consistency within videos, thereby enhancing the accuracy and reliability of video anomaly detection in dynamic videos.

In the video of "D003_05" within the NWPU Campus dataset, puppies crossing the road serves as an example of the appearance anomaly. As shown in Fig. 1, the score curve generated by the DSM exhibits a sharp increase in both abnormal and normal events, indicating a tendency towards false alarms. In contrast, we compare the denoised data with the original data to get a difference and aggregate it with the score function based on the proposed autoregressive denoising score matching mechanism, compensating for the appearance gap via enhancing the perception of low-level pixel details. The results of the ADSM illustrate that the score curve experiences a rapid increase when the puppies

*These authors contributed equally to this work.
†Corresponding author.

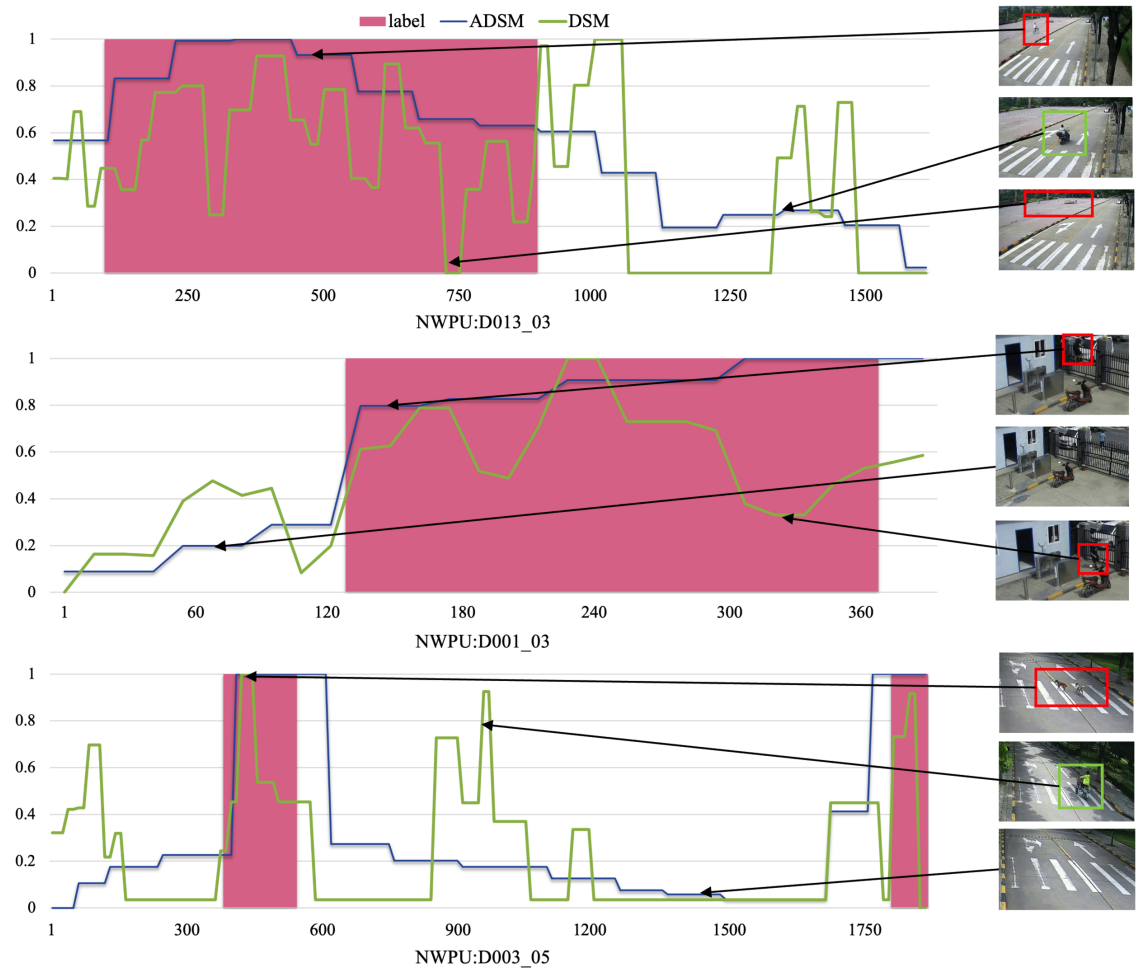


Figure 1. A diagram of the visualization of score curves on the NWPU Campus dataset. "DSM" is the original denoising score matching method without the combination of the autoregressive strategy and the characteristics of video anomalies. "ADSM" is our final autoregressive denoising score matching method. A higher score represents a higher probability of anomaly. Best viewed in color.

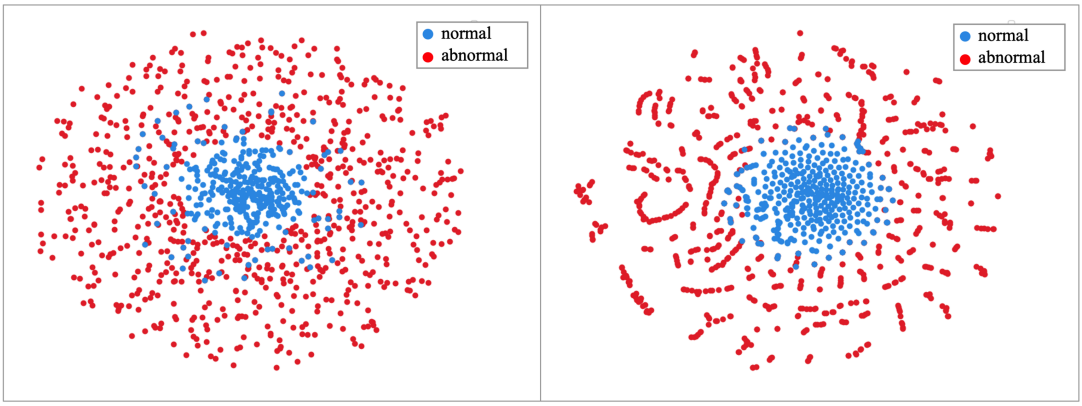
are observed crossing the road, while maintaining a stable, lower value in other normal events. Therefore, our method better identifies appearance anomalies without generating false alarms, equipping the score matching mechanism with enhanced appearance perception.

We further investigate the distribution of normal and abnormal scores generated by two versions of our method (*i.e.*, the DSM and ADSM) via t-SNE [14]. As shown in Fig. 2, results from the DSM reveal that normal and anomaly scores closely cluster together, posing challenges in accurate detection. This corroborates the motivation discussed in Section 1 of the main manuscript that the denoising score matching mechanism is blind to anomalies localized in local modes. Conversely, when utilizing our proposed ADSM, the combination of characteristics of video anomalies and the autoregressive denoising score matching mechanism leads to a more compact intra-class distribution and a more spread-out inter-class distribution, facilitating a clearer distinction between normal and abnormal instances. This observation underscores the fundamental principle behind the efficacy of our method in effectively separating anomalies located in local modes.

2. Additional Results

2.1. Model Size

Our noise-conditioned score transformer (NCST) leverages a series of NCST blocks to establish a flexible architecture, allowing for varied hidden dimensions and multi-head attention layers. Following ViT [4], we identify three distinct configurations of our model, each characterized by different parameter counts and floating-point operations. Moreover, we present their respective performance on the ShanghaiTech dataset, evaluating both micro and macro scores.



DSM ADSM

Figure 2. t-SNE visualization of the distribution of normal scores and abnormal scores output by our model. "DSM" is the original denoising score matching method without the combination of the autoregressive strategy and the characteristics of video anomalies. "ADSM" is our final autoregressive denoising score matching method. Best viewed in color.

As shown in Tab. 1, the NCST-S model has a relatively small amount of parameters and calculations, greatly saving computing resources with a slight loss of performance. The NCST-B model (the setting presented in our main manuscript) has four times the number of parameters and calculations compared with the NCST-S model, achieving an average improvement of 3.5% in terms of the micro and macro scores on the ShanghaiTech dataset. When we continue to increase the model size and get the NCST-L model, there is a slight decrease of 1.8% (2.4%) on the ShanghaiTech dataset in terms of the micro (macro) score. We analyze that the current size of the dataset may not be sufficient to fully utilize the capabilities of the this large model. This still underscores the robust scalability exhibited by our model.

2.2. Patch Size

The patch size typically plays a significant role in model performance. In our main manuscript, we adhere to the standard practice in ViT [4] by utilizing 16×16 as the patch size for our model. We conduct an analysis of the AUC variation on the ShanghaiTech dataset across different patch sizes, in Tab. 2. The model configuration employed is NCST-B, as outlined in Tab. 1, which aligns with the settings presented in Section 4.1 of the main manuscript. Our method implements an autoregressive denoising score matching based on our NCST, in which we develop a patch-wise objective function and propose a motion weighting strategy to guide our model focused on the motion consistency of the input video sequences. As shown in Tab. 2, initially there is an improvement of 2.4% (3.3%) on the ShanghaiTech dataset in terms of the micro (macro) score when we increase the patch size from 8 to 16. The larger patch size contains more complete objects in the input video sequences, which improved the model via enhancing the understanding of more complete spatiotemporal information. However, when we continue to increase the patch size to 32, the excessive patch size leads to performance degradation of 7.6% (8.9%) on the ShanghaiTech dataset in terms of the micro (macro) score. We analyze that the imbalance between static backgrounds and dynamic objects [12] in the VAD dataset may attribute to the insufficient performance. The optimal balance between model capabilities and spatiotemporal information is achieved with the middle value of 16 patch size, which is employed in our main manuscript, delivering the best performance on the ShanghaiTech dataset with superior micro and macro scores.

2.3. Number of Input Frames

The number of input frames usually has an effect on the model performance. In our experiments, we just follow the common setting in action recognition [5, 15, 18] and use 8 frames as the input for our model. We make an analysis of the variation of AUC on the ShanghaiTech dataset for different numbers of input frames in Tab. 3. The model setting we used is NCST-B described in Tab. 1, which is also the setting we report in Section 4.1 of the main manuscript. It can be seen that the performance of our model is not over-sensitive to this hyperparameter. When the number of input frames decreases, the difference between the video frames also decreases, which reduces the difficulty of modeling while limiting the spatiotemporal information of the input video sequences. Conversely, when the number of input frames increases, the model is more likely to perceive stronger spatiotemporal relationships in input video sequences while the difficulty of modeling also increases. Both the increase and decrease of the performance obtained by our method do not exceed 2.5% on the ShanghaiTech

Table 1. AUCs (%) obtained by our method for different configurations of the proposed NCST model on the ShanghaiTech dataset.

Model	Layer numbers	Hidden size	Heads	Params (M)	FLOPs (G)	ShanghaiTech	
						Micro	Macro
NCST-S	12	384	6	33.1	180.4	81.1	89.6
NCST-B	12	768	12	130.7	712.3	**84.5**	**93.2**
NCST-L	24	1024	16	458.3	2511.1	82.7	90.8

Table 2. AUCs (%) obtained by our method for different patch sizes on the ShanghaiTech dataset.

Patch size	8	16	32
Micro	82.1	**84.5**	76.9
Macro	89.9	**93.2**	84.3

Table 3. AUCs (%) obtained by our method for different numbers of input frames on the ShanghaiTech dataset.

Frame	4	8	12	16
Micro	82.1	**84.5**	83.9	82.8
Macro	89.9	**93.2**	92.8	91.1

dataset in terms of the micro and macro scores. The middle value of 8 used in our main manuscript achieves the best performance on the ShanghaiTech dataset in terms of the micro and macro scores, which means a better balance between the capabilities of the model and spatiotemporal information of the input video sequence.

2.4. Motion Weighting Strategy

As introduced in Section 3.4 of the main manuscript, the proposed key frame motion weighting strategy is based on the video codec theory [8], which holds that the first and last key frames of a video clip retain key information about implicit relative relations between appearance and motion. Considering key frames not only balances the computational requirements but also motivates our model to explore and mine the potential spatiotemporal variation relationships therein. We compare our method with several commonly used weighting methods, including calculating average inter-frame difference and rank pooling [6] on the ShanghaiTech dataset in terms of the micro score and macro score. As shown in Tab. 4, compared to computing the difference between frames on average, calculating the difference between key frames yields better results with lower overhead. As a linear representation method, rank pooling may constrain the ability of our transformer-based model to capture non-linear attention in long sequences, limiting its performance compared to our approach. Notably, we verify the input data to ensure the absence of the extreme situation where the first and last frames are exactly the same (resulting in zero weight).

Table 4. AUCs (%) of different motion weighting methods on the ShanghaiTech dataset.

Method	ShanghaiTech	
	Micro	Macro
Average	84.1	92.9
Rank pooling [6]	82.9	90.3
Ours	**84.5**	**93.2**

2.5. Running Speed

As introduced in Section 4.5 of the main manuscript, we perform all the experiments on four NVIDIA GeForce RTX 4090 GPUs with the pytorch framework [11]. Our method is implemented purely on the raw video clip without an extra feature extractor. Furthermore, we utilize the powerful generative model only for a score rather than high-fidelity images or videos, which greatly simplifies the denoising process. This represents a very small computational cost. The running time of our ADSM is primarily determined by the size of our noise-conditioned score transformer. In our setup described in the main manuscript, our ADSM uses an NCST with 130M parameters and takes less than 20 milliseconds to process an input sequence of 8 frames, which enables video anomaly detection at around 50 FPS. Even considering the additional consumption of the object detection model [3, 17], our method is still able to maintain a speed around 35 FPS, suitable for a real-time monitoring system typically around 30 FPS. Moreover, we provide the inference speed and the corresponding performance of five representative methods HF^2-VAD [9], VABD [7], LLSH [10], FPDM [16], and SSAE [2] on the ShanghaiTech dataset in terms of the micro score. As shown in Tab. 5, our proposed method achieves the best results (84.5%) while maintaining a competitive running speed. Notably, the fastest method VABD [7] uses pre-extracted optical flow that can be slowed down by taking the time of the optical flow extractor into account. We argue that our method not only takes the characteristics of video anomalies into consideration but also maintains a stable inference speed without extra input modality.

Table 5. Running speed (in seconds), frames per second (FPS), and AUCs (%) of different methods on the ShanghaiTech dataset during the inference phase. "$^{\diamond}$" represents methods using pre-processed optical flow.

Model	Time (s)	FPS	Micro (%)
$^{\diamond}$HF^2-VAD [9]	0.0667	15	76.2
$^{\diamond}$VABD [7]	**0.0147**	**68**	78.2
LLSH [10]	0.0392	25.5	77.6
FPDM [16]	0.1282	7.8	78.6
SSAE [2]	0.1096	10.1	80.5
Ours	0.0194	51.5	**84.5**

References

[1] Congqi Cao, Yue Lu, Peng Wang, and Yanning Zhang. A new comprehensive benchmark for semi-supervised video anomaly detection and anticipation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20392–20401, 2023. 1

[2] Congqi Cao, Hanwen Zhang, Yue Lu, Peng Wang, and Yanning Zhang. Scene-dependent prediction in latent space for video anomaly detection and anticipation. *IEEE transactions on pattern analysis and machine intelligence*, 2024. 4, 5

[3] MMTracking Contributors. Mmtracking: Openmmlab video perception toolbox and benchmark. In *https://github.com/open-mmlab/mmtracking*, 2020. 4

[4] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 3

[6] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5378–5387, 2015. 4

[7] Jing Li, Qingwang Huang, Yingjun Du, Xiantong Zhen, Shengyong Chen, and Ling Shao. Variational abnormal behavior detection with motion consistency. *IEEE Transactions on Image Processing*, 31:275–286, 2021. 4, 5

[8] Guozhu Liu and Junming Zhao. Key frame extraction from mpeg video stream. In *2010 Third International Symposium on Information Processing*, pages 423–427. IEEE, 2010. 4

[9] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13588–13597, 2021. 4, 5

[10] Yue Lu, Congqi Cao, Yifan Zhang, and Yanning Zhang. Learnable locality-sensitive hashing for video anomaly detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(2):963–976, 2022. 4, 5

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 4

[12] Bharathkumar Ramachandra, Michael J Jones, and Ranga Raju Vatsavai. A survey of single-scene video anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2293–2312, 2020. 1, 3

[13] Shengyang Sun and Xiaojin Gong. Hierarchical semantic contrast for scene-aware video anomaly detection. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 22846–22856, 2023. 1

[14] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 2

[15] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018. 3

[16] Cheng Yan, Shiyu Zhang, Yang Liu, Guansong Pang, and Wenjun Wang. Feature prediction diffusion model for video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5527–5537, 2023. 4, 5

[17] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022. 4

[18] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 803–818, 2018. 3