PREDICTIVE TRANSITIONS FOR VIDEO ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We derive a direct anomaly signal from a representation learned to condition future-frame generation, using a state–transition framework for normal-only video anomaly detection. Current appearance and ordered frame and feature changes jointly condition a diffusion Transformer. After generative training, we freeze the encoders and fit a small head to predict the next local frame change from transition features. Matched representation probes on Ped2, Avenue, and ShanghaiTech show that learned transition features outperform raw frame-change history for anomaly detection. Interventions reveal dependence on transition content and temporal order, with clearer temporal-order effects on Ped2 and Avenue. Together, the probes and conditioning ablations show that transition features can support anomaly detection even when hybrid conditioning does not improve frame-error AUC. The change score can be computed from the learned encoders and prediction head without iterative frame generation.

1 INTRODUCTION

Video anomaly detection (VAD) identifies events that depart from normal scene behavior. Because anomalous events are diverse and difficult to collect, normal-only methods learn from normal videos and identify deviations at test time. Reconstruction and future-frame prediction follow this principle by comparing reconstructed or predicted frames with observations (Hasan et al., 2016; Liu et al., 2018).

The usefulness of prediction error depends on both the prediction target and the information used to predict it. Memory- and prototype-based methods guide reconstruction and prediction with representative normal patterns (Gong et al., 2019; Park et al., 2020; Lv et al., 2021). Appearance–motion learning and temporal auxiliary tasks capture how scenes change over time (Cai et al., 2021; Georgescu et al., 2021). Yet a single frame prediction error combines discrepancies in appearance and temporal change; it does not directly assess whether the next change follows the pattern of preceding changes. Predicting the next change directly from the representation of preceding changes provides a complementary anomaly signal.

We obtain this signal with a state–transition framework. The state representation encodes current appearance, while the transition representation encodes ordered changes in frames and their features. Both representations condition a diffusion Transformer that predicts the next frame. We then train a future-change prediction head on frozen transition features to predict the next local frame change. The resulting error measures deviations in future local changes, providing a change-based anomaly score alongside frame prediction error.

We separately evaluate transition features as conditions for frame generation and as inputs to future-change prediction. Matched representation probes fit the same change-head architecture to state features, the last feature difference, raw frame-change history, and learned transition features, with the encoders frozen. Inference-time interventions then test whether the trained model uses transition content, spatial alignment, and temporal order.

Across Ped2, Avenue, and ShanghaiTech, learned transition features outperform raw frame-change history under unsmoothed scoring. Temporal-order effects are clearer on Ped2 and Avenue, while the effect of hybrid conditioning on frame-error-based detection varies across datasets. Thus, frame-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

error ablations alone do not fully characterize the anomaly information in a conditioning representation. Our contributions are:

- A state–transition framework for studying how current appearance and ordered changes contribute to conditional future-frame prediction.
- A future-change prediction head that provides a direct anomaly signal from frozen transition features without iterative frame generation.
- Matched representation probes and transition interventions that distinguish anomaly information in transition features from their contribution to frame-error-based detection.

2 Related Work

2.1 Video Anomaly Detection

Reconstruction and prediction. Normal-only VAD has developed from models of local dynamics and sparse representations to learned reconstruction and prediction (Mahadevan et al., 2010; Lu et al., 2013; Luo et al., 2017; Hasan et al., 2016). Future-frame prediction detects deviations from learned normal patterns by comparing predicted and observed frames (Liu et al., 2018). MemAE, MNAD, and prototype networks use representative normal patterns to guide reconstruction or prediction (Gong et al., 2019; Park et al., 2020; Lv et al., 2021). Event restoration and masked-autoencoder methods introduce alternative prediction tasks, while feature-matching methods compare observations with representations of normal examples (Yang et al., 2023; Ristea et al., 2024; Ahn et al., 2024).

Motion and temporal representations. Separating visual content from motion is an established approach to video prediction (Villegas et al., 2017; Denton & Birodkar, 2017). In VAD, appearance–motion consistency, temporal direction, and spatio-temporal order provide learning signals beyond individual-frame appearance (Cai et al., 2021; Georgescu et al., 2021; Wang et al., 2022). Local-pattern modeling also relates spatial representations to temporal context (Jiang, 2025).

2.2 Conditional Generative Models for VAD

Denoising diffusion models generate samples by reversing a progressive corruption process, while related flow-matching formulations learn vector fields connecting noise and data (Ho et al., 2020; Lipman et al., 2023). Conditional generation guides synthesis using input information, supporting tasks such as image-to-image translation (Rombach et al., 2022; Saharia et al., 2022). Diffusion Transformers provide token-based architectures for generative modeling (Peebles & Xie, 2023).

Diffusion-based VAD has explored prediction in both feature and frame spaces. FPDM uses two diffusion modules to predict and refine future-frame features, with successive stages emphasizing motion and appearance (Yan et al., 2023). MA-PDM instead performs patch-based frame prediction conditioned on appearance information and temporal frame differences; its anomaly score combines future-frame prediction error with appearance reconstruction error (Zhou et al., 2025). These methods derive anomaly scores from predicted or reconstructed frames and features. We additionally predict the next local change from the transition representation that conditions generation and use its error as an anomaly score.

3 Method

Figure 1 presents the proposed framework. State and transition features condition future-frame prediction, while a change head predicts the next local frame change from transition features. The two prediction errors provide separate anomaly scores.

3.1 Problem Formulation

We formulate normal-only VAD through future-frame and future-change prediction. Given the observed history $\mathcal{F} = (F^1, \ldots, F^N)$, the model predicts the next frame F^{N+1} and its local change

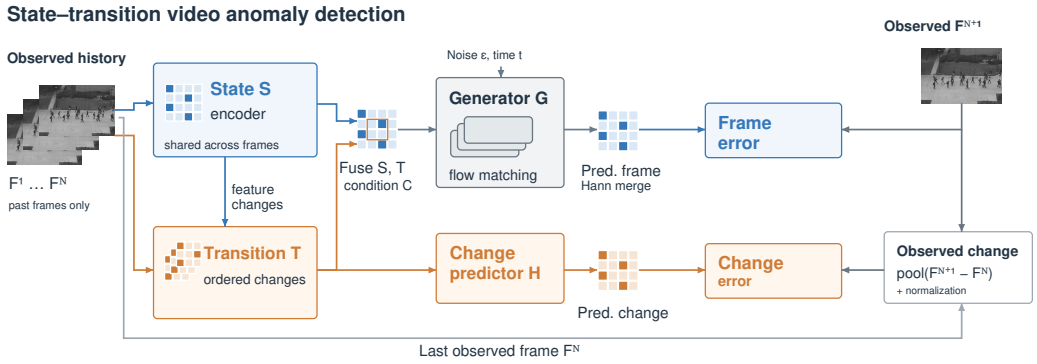


Figure 1: The proposed state–transition framework. State and transition features jointly condition the frame predictor, while the change head predicts the next local frame change from T. Comparing both predictions with the next observation yields frame error and change error. Equation 7 combines the two scores. Feature grids and predictions are schematic.

relative to F^N. Model parameters are learned from normal videos. At test time, the frame predictor starts from Gaussian noise and uses conditions computed from the history; the change head predicts directly from the transition representation. Once F^{N+1} is observed, the two prediction errors measure departures from the learned patterns. Superscripts index frames, while t denotes continuous generative time.

3.2 Diffusion Model with State and Transition Conditions

We distinguish the current visual state from ordered changes in the observed history. A shared frame encoder yields features for each observed frame. The transition encoder combines consecutive frame differences with changes in the corresponding features:

$$\begin{aligned} S^i &= \mathrm{Encoder}_{\mathrm{state}}(F^i), \\ \Delta F^i = F^{i+1} - F^i, &\qquad \Delta S^i = S^{i+1} - S^i, \quad i = 1, \dots, N-1, \\ T &= \mathrm{Encoder}_{\mathrm{transition}}\left((\Delta F^i, \Delta S^i)_{i=1}^{N-1} \right). \end{aligned} \tag{1}$$

The latest frame supplies the state $S = S^N$, and the fused condition is $C = \mathrm{Fuse}(S, T)$. Signed frame differences capture observed changes without a separate optical-flow estimator.

We use a conditional flow-matching formulation (Lipman et al., 2023) with a diffusion Transformer that predicts the clean frame (Li & He, 2025). Its output defines the velocity used to generate a frame from noise. For $t \in [0, 1]$, the path between noise and the target frame is

$$Z_t = (1-t)\epsilon + tF^{N+1}, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

Thus, Z_0 is Gaussian noise and Z_1 is the clean target. Adding noise moves toward $t = 0$; generation follows the learned velocity toward $t = 1$. We use frame-level notation here and describe the local token implementation in Appendix A.1.

Given C, the clean-frame predictor G_θ induces the velocity field

$$\begin{aligned} V_\theta(Z_t, t, C) &= \frac{G_\theta(Z_t, t, C) - Z_t}{1-t}, \\ V_t^* &= \frac{F^{N+1} - Z_t}{1-t} = F^{N+1} - \epsilon, \qquad t < 1. \end{aligned} \tag{3}$$

We optimize the conditional frame predictor with

$$\mathcal{L}_{\mathrm{frame}} = \mathbb{E}_{\mathcal{F}, F^{N+1}, t, \epsilon} \left[\mathrm{MSE}(V_\theta(Z_t, t, C), V_t^*) \right]. \tag{4}$$

Time sampling and numerical stabilization are specified in Appendix B.1.

3.3 FUTURE-CHANGE PREDICTION

Frame prediction error measures discrepancies between predicted and observed frames. To assess temporal change directly, we train a future-change prediction head, referred to as the *change head*, to predict $\widehat{D}^{N+1} = H_\psi(T)$. The same small multilayer perceptron is applied at every spatial location.

The target is the observed local change $D^{N+1} = \Phi(F^{N+1} - F^N)$. The operation Φ pools frame differences locally and standardizes them using statistics from normal clips used to train the head, retaining both signed changes and change magnitudes. Appendix A.2 specifies the target construction and training loss.

Two-stage training. First, the state encoder, transition encoder, and frame predictor are jointly optimized with Equation 4. We then freeze the encoders and train only the change head on normal clips. Its L1 loss balances changing and relatively static regions so that static background does not dominate training. The head checkpoint is selected by validation loss on normal clips. Fitting the same head architecture to other frozen inputs defines the representation-probing experiments in Section 4.2.

3.4 ARCHITECTURE

Condition encoders. A shared convolutional encoder with residual blocks maps each observed frame to a spatial feature grid. Its latest output forms S. The transition encoder combines consecutive frame and feature differences. Temporal positional embeddings encode their order; temporal attention aggregates them at each location, followed by spatial attention across locations (Vaswani et al., 2017). State and transition features are projected to a common dimension and fused into spatially aligned condition tokens.

Diffusion Transformer. The frame predictor uses a diffusion Transformer architecture (Peebles & Xie, 2023). Following Vision Transformers (Dosovitskiy et al., 2021), it maps the noisy frame to image tokens with positional embeddings. Initial blocks jointly process image and condition tokens; later blocks refine the image tokens and project them back to clean frame values. A generative-time embedding modulates the blocks, and Equation 3 converts the output to a velocity. Generation takes place in frame space, with tokens serving as the internal representation. The model processes aligned local token groups and combines their outputs into a frame; spatial implementation details are given in Appendix A.1.

3.5 INFERENCE AND ANOMALY SCORING

For the frame-generation path, the observed history determines S, T, and C, which remain fixed during generation. Starting from Gaussian noise, we integrate the learned velocity with K Euler steps. With $t_k = k/K$, the update in frame-level notation is

$$Z_{t_{k+1}} = Z_{t_k} + \frac{1}{K} V_\theta(Z_{t_k}, t_k, C), \qquad k = 0, \ldots, K-1. \tag{5}$$

The generated spatial outputs are combined into $\widehat{F}^{N+1}$. Separately, the change head predicts $\widehat{D}^{N+1} = H_\psi(T)$ without iterative generation. Once F^{N+1} is observed, we compute frame prediction error and change prediction error:

$$\begin{aligned} A^{N+1}_{\text{frame}} &= \text{MSE}(F^{N+1}, \widehat{F}^{N+1}), \\ A^{N+1}_{\text{change}} &= \text{Agg}\Big(\Big|D^{N+1} - \widehat{D}^{N+1}\Big|\Big). \end{aligned} \tag{6}$$

We refer to these scores as *frame error* and *change error*. The operation Agg combines local absolute errors into one score per frame (Appendix B.2). The next observation is used only to compute the errors, not as a prediction input.

The two scores can also be combined:

$$A_{\text{fused}} = (1-\lambda)\mathcal{R}(A_{\text{frame}}) + \lambda \mathcal{R}(A_{\text{change}}), \qquad 0 \le \lambda \le 1. \tag{7}$$

Here, λ controls the contribution of change error to the combined score. The operation $\mathcal{R}$ applies a score transformation and per-video min–max normalization before combination; a temporal filter is then applied to the combined score. Normalization uses the complete video, making this an offline score. Appendix B.2 lists the transformations, filters, and weights, and Section 4.1 describes their selection.

Equation 7 defines three scoring paths. For $\lambda = 0$, only frame error is used. For $\lambda = 1$, the *change-only* path uses only the learned encoders and change head. For $0 < \lambda < 1$, both predictions are required.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate the proposed method on three standard VAD benchmarks using their normal training videos:

- **UCSD Ped2** (Mahadevan et al., 2010) contains 16 training and 12 test videos of a pedestrian scene.
- **Avenue** (Lu et al., 2013) contains 16 training and 21 test videos, with anomalies such as running and throwing objects.
- **ShanghaiTech** (Luo et al., 2017) contains 330 training and 107 test videos from 13 scenes, covering a wider range of backgrounds and activities.

Evaluation metric. Our results use frame-level ROC-AUC averaged equally across videos containing both normal and anomalous evaluated frames (*video-macro AUC*). Scoring begins after the six-frame history. Videos with only one label in the evaluated interval are excluded from the macro average, leaving 8/12, 19/21, and 106/107 test videos for Ped2, Avenue, and ShanghaiTech, respectively. This differs from pooled AUC over all frames. Published results retain their source protocols and are presented separately.

Training and inference. Frames are resized to 256×256, and each input clip contains six observed frames. Frame prediction uses eight Euler steps at inference. Appendix B.1 lists the architecture, optimizer settings, dataset-specific training schedules, and random seeds.

Analysis protocol and score selection. Representation probes (Table 1) and transition interventions (Table 2) use unsmoothed scores. Representation probes use three backbone seeds on Ped2 and Avenue and one on ShanghaiTech; the intervention table uses one checkpoint per dataset. Tables 3–5 use smoothed scores, obtained by per-video min–max normalization, the combination in Equation 7, and a temporal median filter. We select λ and the filter length by test AUC (Appendix B.2).

4.2 Anomaly Information in Frozen Representations

We first evaluate transition features learned for frame generation as inputs to direct change-based anomaly detection. Table 1 compares four inputs to the same change-head design: current state features, the last feature difference, raw frame-change history, and learned transition features. All heads are trained on normal clips to predict the same local future-change target.

Learned transition features outperform raw frame-change history by 1.45, 0.92, and 0.80 percentage points on Ped2, Avenue, and ShanghaiTech, respectively. They also outperform the last feature difference on all three datasets. State features remain slightly stronger on ShanghaiTech. These results support direct anomaly scoring from learned transition features.

4.3 Transition Content and Temporal Order

We next test how the trained model uses transition information. The two panels of Table 2 evaluate interventions at different stages: frame-error comparisons perturb the transition condition supplied to generation, while change-error comparisons perturb temporal order before transition encoding.

Table 1: Frozen-representation probes, video-macro ROC-AUC (%), without temporal smoothing. Each head predicts the same future-change target. Learned features come from frozen hybrid models. Ped2 and Avenue report mean ± sample standard deviation over three backbone seeds; ShanghaiTech uses one.

Head input	Ped2	Avenue	ShanghaiTech
Current state	92.54 ± 0.90	82.55 ± 0.17	79.91
Last feature difference	94.50 ± 0.41	83.13 ± 0.05	79.36
Raw frame-change history	94.23 ± 0.08	83.52 ± 0.07	78.68
Learned transition	95.68 ± 0.20	84.44 ± 0.08	79.48

Table 2: Inference-time transition interventions, video-macro ROC-AUC (%). One hybrid checkpoint per dataset; no retraining or temporal smoothing. Frame-error comparisons keep state features and sampling noise fixed. Change-error comparisons alter the order of the change history before encoding.

Score	Setting	Ped2	Avenue	ShanghaiTech
Frame error	Intact transition	91.10	84.71	81.16
	Zero transition	76.90	84.68	78.59
	Transition from another clip	76.20	84.19	78.39
	Spatial permutation	79.57	84.37	80.23
Change error	Intact order	95.75	84.35	79.48
	Reversed order	93.33	81.26	78.76
	Cyclic shift	93.91	81.60	79.19

Removing the transition condition, replacing it with another clip's condition, or permuting its spatial locations lowers frame-error AUC by 11.53–14.90 points on Ped2, 0.93–2.77 points on ShanghaiTech, and at most 0.52 points on Avenue. The trained frame predictor therefore relies on transition information, most strongly on Ped2.

For change error, reversing or cyclically shifting the change history preserves the state frame and prediction target. Both perturbations lower video-macro AUC on all three datasets, by 1.84–3.09 points on Ped2 and Avenue and 0.29–0.72 points on ShanghaiTech. Together with the probes, these results support the role of ordered history in change-based anomaly detection.

4.4 Conditioning and Scoring Paths

We examine two roles of transition features: as a condition for frame prediction and as input to change prediction. Table 3 compares the frame errors of separately trained conditioning models. Table 4 then compares scoring paths from the same hybrid checkpoint.

Hybrid conditioning improves frame-error AUC on Ped2; state-only conditioning is stronger on Avenue and comparable on ShanghaiTech. Transition features nevertheless support direct change-based detection on all three datasets (Table 1).

Under matched processing within each dataset, combining the two errors improves AUC on Avenue and, by a smaller amount, ShanghaiTech. Ped2 instead selects the change-only endpoint, $\lambda = 1$. The benefit of combining the scores is therefore dataset-dependent.

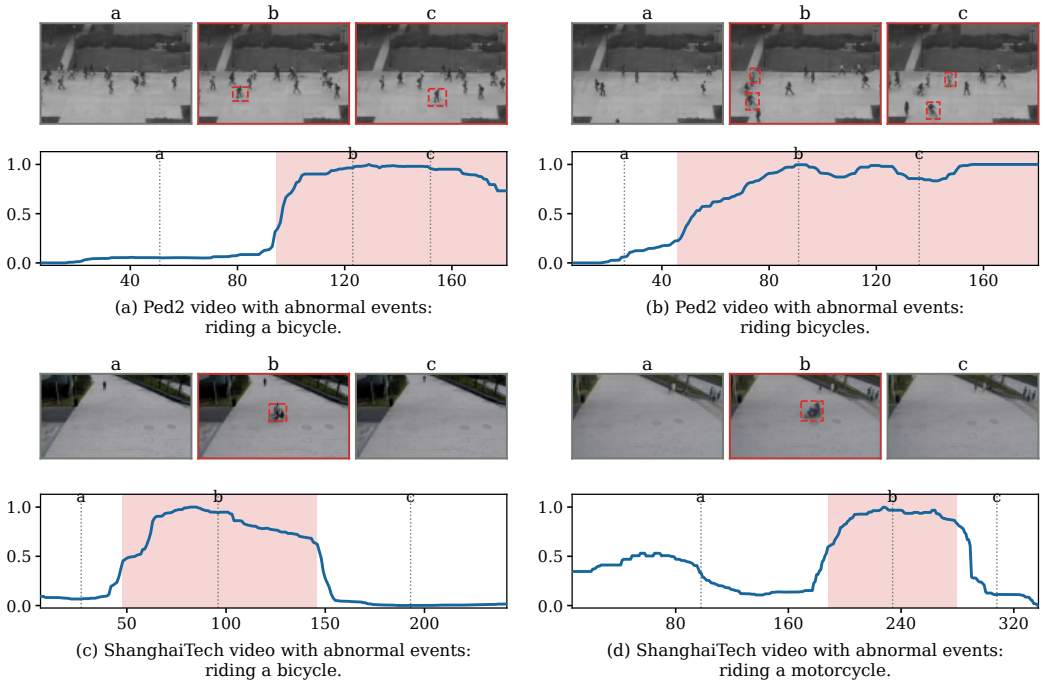
Qualitative examples. Figure 2 shows the selected score over time for two Ped2 and two ShanghaiTech videos. Scores rise during annotated bicycle and motorcycle events; the ShanghaiTech scores also decline afterward.

Table 3: Conditioning ablation, video-macro ROC-AUC (%). Separately trained models are evaluated using smoothed frame error.

Conditioning	Ped2	Avenue	ShanghaiTech
State	89.9	87.9	82.5
Transition	92.5	84.2	78.9
State + Transition	94.4	84.7	82.4

Table 4: Scoring-path comparison with smoothed scores, video-macro ROC-AUC (%). One hybrid model and its change head supply both errors per dataset. Processing is fixed across rows within each dataset; only λ varies. The selected score uses the weights in Table 8; Ped2 selects change error alone.

Configuration	Ped2	Avenue	ShanghaiTech
Frame error	94.36	84.63	82.39
Change error	99.66	85.24	80.76
Selected score	99.66	86.89	82.94



(a) Ped2 video with abnormal events: riding a bicycle.

(b) Ped2 video with abnormal events: riding bicycles.

(c) ShanghaiTech video with abnormal events: riding a bicycle.

(d) ShanghaiTech video with abnormal events: riding a motorcycle.

Figure 2: Selected detection examples from Ped2 and ShanghaiTech using the selected score in Table 4. Shading and red boxes mark ground-truth anomalies; a/b/c connect thumbnails to their frame positions.

4.5 Comparison with Published Methods

Table 5 places the selected scores alongside representative reconstruction and prediction methods, including FPDM and MA-PDM.

Table 5: Published VAD results, ROC-AUC (%). Prior-work values are as reported in the original papers. Our row reports video-macro AUC with smoothed scores.

Method	Venue	Ped2	Avenue	ShanghaiTech
FFP (Liu et al., 2018)	CVPR18	95.4	84.9	72.8
MemAE (Gong et al., 2019)	ICCV19	94.1	83.3	71.2
MNAD-R (Park et al., 2020)	CVPR20	90.2	82.8	69.8
MPN (Lv et al., 2021)	CVPR21	96.9	89.5	73.8
AMMC-Net (Cai et al., 2021)	AAAI21	96.6	86.6	73.7
DLAN-AC (Yang et al., 2022)	ECCV22	97.6	89.9	74.7
USTN-DSC (Yang et al., 2023)	CVPR23	98.1	89.9	73.8
FPDM (Yan et al., 2023)	ICCV23	–	90.1	78.6
MA-PDM (Zhou et al., 2025)	AAAI25	98.6	91.3	79.2
Ours	–	99.66	86.89	82.94

5 Conclusion

We studied transition representations learned as conditions for future-frame generation. Matched probes show that these frozen features yield higher change-based detection AUC than raw frame-change history. Interventions support dependence on transition content and temporal order, with clearer temporal-order effects on Ped2 and Avenue. The contribution of transition conditioning to frame-error-based detection is dataset-dependent, showing why frame-error ablations alone do not fully characterize the representation's usefulness. The change-only path makes this information available for detection without iterative frame generation.

Limitations

ShanghaiTech uses one backbone seed, and its temporal-order effects remain uncertain. End-to-end inference efficiency remains to be evaluated. We have not compared against encoders trained directly for change prediction without generative training; the necessity of generative training is therefore unresolved.

AI Use Statement

Generative AI tools assisted with literature search, experimental planning, implementation support, result interpretation, figure preparation, and manuscript drafting and editing. The authors are responsible for verifying the methods, analyses, references, and final content.

References

Sunghyun Ahn, Youngwan Jo, Kijung Lee, and Sanghyun Park. Videopatchcore: An effective method to memorize normality for video anomaly detection. In *Proceedings of the Asian Conference on Computer Vision*, pp. 2179–2195, 2024.

Ruichu Cai, Hao Zhang, Wen Liu, Shenghua Gao, and Zhifeng Hao. Appearance-motion memory consistency network for video anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):938–946, 2021. doi: 10.1609/aaai.v35i2.16177.

Emily L. Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1705.10915.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Mariana-Iuliana Georgescu, Antonio Barbalau, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. Anomaly detection in video via self-supervised and multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12742–12752, 2021. URL https://arxiv.org/abs/2011.07491.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1705–1714, 2019.

Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, and Larry S. Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–742, 2016. URL https://arxiv.org/abs/1604.04574.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2006.11239.

Yalong Jiang. Local patterns generalize better for novel anomalies. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4ua4wyAQLm.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. URL https://arxiv.org/abs/2511.13720.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection – a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6536–6545, 2018.

Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 FPS in MATLAB. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2720–2727, 2013.

Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked RNN framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 341–349, 2017.

Hui Lv, Chen Chen, Zhen Cui, Chunyan Xu, Yong Li, and Jian Yang. Learning normal dynamics in videos with meta prototype network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15425–15434, 2021. URL https://arxiv.org/abs/2104.06689.

Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1975–1981, 2010.

Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14372–14381, 2020.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Nicolae-Catalin Ristea, Florinel-Alin Croitoru, Radu Tudor Ionescu, Marius Popescu, Fahad Shahbaz Khan, and Mubarak Shah. Self-distilled masked auto-encoders are efficient video anomaly detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2306.12041.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://arxiv.org/abs/2112.10752.

Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. URL https://arxiv.org/abs/2111.05826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1706.08033.

Guodong Wang, Yunhong Wang, Jie Qin, Dongming Zhang, Xiuguo Bao, and Di Huang. Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles. In *European Conference on Computer Vision*, pp. 494–511, 2022. URL https://arxiv.org/abs/2207.10172.

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*, pp. 5527–5537, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Yan_Feature_Prediction_Diffusion_Model_for_Video_Anomaly_Detection_ICCV_2023_paper.html.

Zhiwei Yang, Peng Wu, Jing Liu, and Xiaotao Liu. Dynamic local aggregation network with adaptive clusterer for anomaly detection. In *Computer Vision – ECCV 2022*, pp. 404–421, 2022. doi: 10.1007/978-3-031-19772-7_24.

Zhiwei Yang, Jing Liu, Zhaoyang Wu, Peng Wu, and Xiaotao Liu. Video event restoration based on keyframes for video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14592–14601, 2023. doi: 10.1109/CVPR52729.2023.01402.

Hang Zhou, Jiale Cai, Yuteng Ye, Yonghui Feng, Chenxing Gao, Junqing Yu, Zikai Song, and Wei Yang. Video anomaly detection with motion and appearance guided patch diffusion model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(10):10761–10769, 2025. doi: 10.1609/aaai.v39i10.33169.

A Implementation Details

A.1 Architecture and Local Token Processing

The frame predictor adapts the pixel-space JiT backbone of Li & He (2025) for state–transition conditioning. A bottleneck embedding maps frame values to image tokens, and the generative-time embedding supplies adaptive modulation and residual gates. Table 6 lists the dimensions and block counts.

In the transition encoder, a learned summary token aggregates the ordered changes at each location before spatial attention operates over the condition grid. Temporal and spatial positional embeddings encode order and location.

State and transition grids are computed before selecting spatial regions for the frame predictor. During training, one random window w is selected per clip. Let r_w extract a frame region and q_w its aligned feature region:

$$f_w^{N+1} = r_w(F^{N+1}), \qquad s_w = q_w(S), \qquad \tau_w = q_w(T), \qquad c_w = P_s(s_w) - P_\tau(\tau_w) - p_w. \tag{A.1}$$

Here P_s and P_τ are learned projections and p_w is the aligned positional embedding. This subtraction defines the implemented feature fusion. Each window supplies image tokens and their aligned condition tokens. The path and loss in Equations 2 and 4 are applied to $(f_w^{N+1}, z_{t,w}, c_w)$.

At test time, each overlapping window starts from independent Gaussian noise and follows the Euler update in Equation 5. Generated windows are clipped to the frame range and combined using normalized Hann overlap weights. Thus, frame generation uses local token groups, while the transition encoder processes the full condition grid.

A.2 Future-Change Targets and Training

The target $D^{N+1} = \Phi(F^{N+1} - F^N)$ represents local frame changes on the condition grid. Within each cell, we pool positive and negative differences separately. Two additional components capture change magnitude: average pooling of the channel-mean absolute difference, and max pooling of the channel-maximum absolute difference. For transition feature $\tau_j = T[j]$, the target and prediction are

$$d_j^{N+1} = \mathrm{Norm}_{\mathrm{train}}\big(\mathrm{Pool}_j(F^{N+1} - F^N)\big), \qquad \widehat{d}_j^{N+1} = H_\psi(\tau_j). \tag{A.2}$$

Each target component is standardized using its mean and sample standard deviation over targets from the normal clips used to train the head. These statistics remain fixed during validation and testing. The change head applies the same multilayer perceptron independently at every cell; Table 6 specifies its layers and output dimensions.

A cell is active when the mean of its unstandardized average- and maximum-magnitude components exceeds the threshold in Table 6. Let $\mathcal{A}$ contain active cells and $\mathcal{B}$ the remaining cells in a training batch. We balance their contributions to the L1 loss:

$$e_j = \mathrm{Mean}\left|d_j^{N+1} - \widehat{d}_j^{N+1}\right|, \qquad \mathcal{L}_{\mathrm{change}} = \tfrac{1}{2}\,\mathrm{Mean}_{j\in\mathcal{A}}(e_j) + \tfrac{1}{2}\,\mathrm{Mean}_{j\in\mathcal{B}}(e_j). \tag{A.3}$$

If either group is empty, the loss averages all cell errors.

B Training and Evaluation Settings

B.1 Model and Training Settings

Tables 6 and 7 summarize the architecture, optimization settings, and dataset-specific training schedules.

The velocity-denominator lower bound in Table 6 applies to both predicted and target velocities.

Head initialization seeds are 42 for transition, 43 for state and last feature difference, and 44 for raw frame-change history. The head data-split seed and conditioning-ablation seed are 42.

Table 6: Shared architecture and optimization settings.

Setting	Value
History / frame resolution / frame range	6 / 256×256 / $[-1, 1]$
State widths / feature width	64, 128, 128, 256 / 256
Transition temporal / spatial blocks / heads	2 / 1 / 4
Window size / token patch size / test stride	128 / 16 / 64
Frame Transformer width / blocks / attention heads	768 / 12 / 12
Token bottleneck / joint-conditioning blocks	128 / 4
Backbone optimizer / learning rate / weight decay	AdamW / 10^{-4} / 10^{-5}
Batch size / gradient accumulation / gradient clip	32 / 4 / 1.0
Schedule / warm-up fraction	Cosine / 0.05
Time sampling	$t = \mathrm{sigmoid}(0.8g - 0.8),\ g \sim \mathcal{N}(0, 1)$
Velocity denominator lower bound / Euler steps	10^{-4} / 8
Change head	LayerNorm, Linear(256), SiLU, Linear(128)
Change cell / pooling resolution / output size	16×16 / 4×4 / 128
Change target standard-deviation lower bound	10^{-4}
Active-cell threshold / score top-κ	0.02 / 4
Head optimizer / learning rate / weight decay	AdamW / 10^{-3} / 10^{-4}
Head batch size / feature augmentation	16 / none

Table 7: Dataset-specific training settings and random seeds.

Setting	Ped2	Avenue	ShanghaiTech
Backbone epochs	1,000	300	30
Head-training clips	2,022	4,000	6,000
Head-validation clips	432	1,200	1,500
Maximum head epochs / patience	60 / 10	40 / 8	40 / 8
Hybrid backbone seeds for probing	42, 7, 123	42, 7, 123	42

B.2 Score Aggregation and Post-processing

Frame error is the mean squared difference between the predicted and observed frames after spatial outputs are combined. Using the cell errors e_j defined in Equation A.3, change error adds the average cell error to the average of the largest cell errors:

$$\mathrm{Agg}\Big(\Big|D^{N+1} - \widehat{D}^{N+1}\Big|\Big) = \mathrm{Mean}_j(e_j) + \mathrm{Mean}_{j \in \mathrm{Top}_\kappa(e)}(e_j). \tag{B.1}$$

For the smoothed scores, we transform each error, apply per-video min–max normalization, combine the normalized scores using Equation 7, and then apply a centered median filter with nearest boundary handling. Avenue uses a square-root transformation for both errors; Ped2 and ShanghaiTech use no transformation. Normalization uses the complete video, making this an offline evaluation.

Table 8 gives the selected weights and filter lengths. We select the weights, filter lengths, and transformations by test AUC. Within each dataset, processing is fixed and only λ varies: zero for frame error, one for change error, or the selected weight. Ped2 selects $\lambda = 1$, so its selected score is change error alone.

Table 8: Scoring settings for the smoothed scores. Each dataset applies the same transformation and normalization to both errors before combination; the median filter is applied afterward.

Setting	Ped2	Avenue	ShanghaiTech
Change weight λ	1.000	0.437	0.250
Median filter length	49	17	33