Multimodal Motion Conditioned Diffusion Model for Skeleton-based Video Anomaly Detection
Supplementary Materials

Alessandro Flaborea* Luca Collorone* Guido Maria D'Amely di Melendugno*
Stefano D'Arrigo* Bardh Prenkaj Fabio Galasso
{flaborea,damely,darrigo,prenkaj,galasso}@di.uniroma1.it luca.collorone@uniroma1.it

Sapienza University of Rome, Italy
https://github.com/aleflabo/MoCoDAD

We supplement the main paper by adding an ablation study on several features of the diffusion process, providing further insights into the conditioning strategies of our method, and qualitatively illustrating the inference diffusion process, the generated distribution manifolds and pictograms of input poses with various corrupting types of noise. The following table of contents outlines how the supplementary material is organized.

Contents

A. Diffusion Process

A.1. Background on Diffusion Models

A denoising diffusion probabilistic model (DDPM) [7, 22] exploits two Markov chains: i.e., a *forward process* and a *reverse process*. The forward process $q(x_t|x_{t-1})$ corrupts the data $x = x_0$ gradually adding noise according to a variance schedule $\beta_t \in (0, 1)$ for $t = 1, \dots, T$, transforming any data distribution $q(x_0)$ into a simple prior (e.g., Gaussian). The forward process can be expressed as

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbb{I}) \tag{1}$$

To shift the data distribution $q(x_0)$ toward $q(x_t|x_{t-1})$ in one single step, equation 1 can be reformulated as

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbb{I}) \tag{2}$$

with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$.

The reverse process leans to roll back this degradation. More formally, the reverse process can be formulated as

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\big(x_{t-1}; \mu_\theta(x_t, t), \beta_t\mathbb{I}\big) \tag{3}$$

where $\mu_\theta(x_t, t)$ is a deep neural network that estimates the forward process posterior mean. [7] has shown that one obtains high-quality samples when optimizing the objective

$$\mathcal{L}_{simple} = \mathop{\mathbb{E}}_{t,x_0,\varepsilon}\left[\left|\left|\varepsilon - \varepsilon_\theta\big(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon, t\big)\right|\right|_2^2\right] \tag{4}$$

where $\varepsilon \sim \mathcal{N}(0, \mathbb{I})$ is the noise used to corrupt the sample x_0, and ε_θ is a neural network trained to predict ε.

During inference, the sampling algorithm of [7] is used to iteratively denoise random Gaussian noise $x_T \sim \mathcal{N}(0, \mathbb{I})$, to generate a sample from the learned distribution. [15, 16] have shown that one may condition DDPMs on a signal h by feeding it to the neural network ϵ_θ.

A.2. Diffusive steps

Sec. 3 in the main paper delineates a gradual corruption technique that employs a displacement map in accordance

*Authors contributed equally.

Table 1: AUC-ROC performance variation of MoCoDAD on the number of employed diffusive steps t of the variance scheduler β_t.

Diffusive steps	HR-UBnormal	UBnormal
2	65.0	64.7
5	66.3	65.9
10	**68.4**	**68.3**
25	64.70	64.6
50	64.4	64.4

with a variance scheduler $\beta_t \in (0, 1)$ to corrupt the input. The degree of displacement applied is determined by β_t which follows a schedule based on the parameter t.

To further investigate the relationship between the diffusive steps and performance, we evaluate the impact of varying t on the performance of MoCoDAD. As shown in Table 1, we consider five different steps $t \in \{2, 5, 10, 25, 50\}$. Our results demonstrate that optimal performances occur with 10 diffusive steps while deteriorating for any higher or lower value.

Note that this optimal T value is significantly smaller than those used in other diffusion approaches [7, 11, 14, 18] which require a large number of steps to turn a noisy sample $x_T \in \mathcal{N}(0, I)$ into a semantically significant one.

On the contrary, our model's strong inductive bias towards poses, together with its ability to leverage the invariant relationships and dependencies between intra-pose joints allows it to transform a set of random joint positions $x_T \in \mathcal{N}(0, I)$ into a pose-like structure in just one step, as illustrated in Figure 3. Furthermore, we highlight that a small T can push the model to improve the quality of normal poses while failing to refine abnormal ones. Thereby, employing a $T = 10$ we allow MoCoDAD to foster this trade-off and obtain optimal performances.

Additionally, to highlight the effectiveness of the iterative diffusion process we evaluate the model with $T = 2$, that is, the case where the model only receives either clean or completely corrupted input poses: this means that during inference the model performs the denoising non-iteratively, e.g. in one single step. The resulting model underperforms w.r.t. MoCoDAD, confirming the importance of a multi-step diffusion process.

B. Weaker forms of conditioning

In Table 2, we complement the experiments in Sec. 5.3 of the main paper with an additional discussion on the forms of conditioning. Following the approach proposed by [21], we investigate two aspects: applying an alternative sampling strategy and using a different corruption function instead of the Gaussian one. We also evaluate the effectiveness of MoCoDAD in the absence of conditioning past motion frames.

Table 2: Impact of different noise distributions and sampling strategies on performance in terms of AUC-ROC. MoCo refers to Motion Condition; T represents the diffusion step at which samples are completely corrupted; γ represents the step up to which samples are corrupted during inference. The last row illustrates our proposed method, MoCoDAD.

γ/T	Corruption	MoCo	HR-UBnormal	UBnormal
3/10	Simplex	×	53.0	52.0
3/10	Gaussian	×	57.4	57.3
10/10	Gaussian	×	55.0	54.1
10/10	Gaussian	✓	**68.4**	**68.3**

Regarding the alternative sampling strategy, we train our diffusion model to denoise a corrupted sample x_t, where $t \in \{1, \dots, T\}$ and $T = 10$, while, during inference, we perform sampling starting from partially corrupted samples x_γ where $\gamma < T$. The partially corrupted signal acts as a weaker form of conditioning, i.e., generating by denoising the signal. Hence, the reverse diffusion process does not need to be conditioned on past frames.

The first column of Table 2 refers to the timesteps used at inference time (γ) and the ones used at training (T). Following [21], we set γ to be equal to a third of T. When the denoising process begins with a partially corrupted image (first and second rows), the results degrade to 52 and 57.35, respectively. We explain this since, even in the absence of prior motion, the starting point of the denoising process is more similar to the target signal, reducing the reconstruction error for both normal and abnormal samples.

We investigate this intuition by comparing two different noise distributions to randomly corrupt the poses, namely Gaussian and Simplex noise [13]. Fig. 2 compares the joint displacement at $t \in \{3, 6, 9\}$ for both these noise distributions. We see that Gaussian corrupts the input motion more since every joint is translated with a random intensity, whereas, Simplex acts as a weaker perturber maintaining a significant amount of information from the original motion. This reflects in performance. Table 2 shows that adding Simplex noise is not effective with motion sequences, deteriorating the overall performances to 52 (see row 1).

Next, we consider generating future motions without conditioning on the past. In the absence of conditioning past frames, the model is expected to provide samples from the learned training normal distribution. Therefore, it still makes sense to consider this approach for anomaly detection by comparing similarities of generated and true futures. In fact, the generated future frames will be normal, more similar to normal true futures, and less similar to abnormal

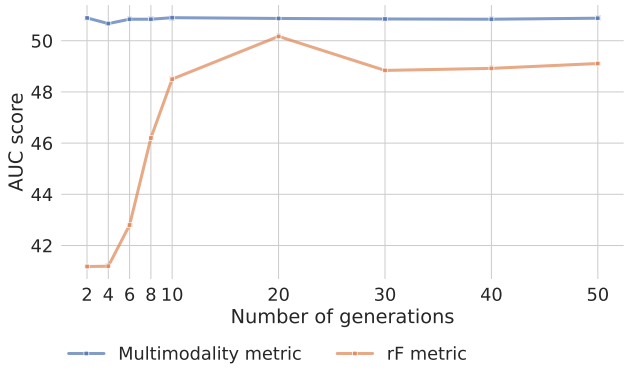

Figure 1: Anomaly detection performance trend when assuming a diversity metric as the anomaly score. It is worth noting that the rF metric yields results that are below the chance level.

true futures. Note, however, that missing to condition on the past will result in general futures unrelated to the specific past, just normal. The results in Table 2 support this observation, i.e. the performance reduces to 54.11, close to the chance level (50%).

To sum up, neither the Gaussian nor the Simplex noise provide comparable performance with MoCoDAD confirming the need for a conditioning signal to govern the diffusion process.

C. MoCoDAD Algorithms

In this section, we outline the algorithms designed for both the training and inference phases of our proposed model (cf. Sec. 3.2 of the main paper). In algorithms 1 and 2, we employ the following notation: $\bar{\cdot}$ denotes the objects that are encoded in a latent space, whereas $\hat{\cdot}$ signifies the predictions of our model.

Train. In Alg. 1 we describe the training process on a single sequence of poses $X^{1:N}$. The algorithm only requires the input sequence, the current timestep t, the parameters λ_1 and λ_2 governing the importance of the two losses, and the MoCoDAD modules introduced in Sec. 3.3 of the main paper.

Inference. Alg. 2 depicts how our proposed method assigns the anomaly score to each frame of a video. For readability purposes, we only examine the case of a single window $\mathcal{W}$, which encompasses the frames $f_1, f_2, ..., f_N$. We then adopt a sliding window procedure to analyze each video so that Alg. 2 can be further extended to assess all the frames of a video. First, we extract the poses of all the subjects whose motion lies in all the frames of $\mathcal{W}$, resulting in the set $\mathcal{A}$. Then, starting from random noise ε, we iteratively leverage MoCoDAD to draw m possible futures in T steps, which we subsequently compare with the GT future to distill m scores for each sample $X_a^{1:N}$ (collected in the set $\mathcal{G}$). As discussed in Sec. 3.2 of the main paper, we then aggregate these scores in a single value ($\mathcal{H}_a$, which we interpret as the anomaly score of the subject a for the frames in $\mathcal{W}$). Note that, when considering multiple overlapping time windows

$$\mathcal{W}^{(1)}[f_1 : f_N], \mathcal{W}^{(2)}[f_2 : f_{N+1}], ..., \mathcal{W}^{(N)}[f_N : f_{2N-1}],$$

$\mathcal{H}_a$ is computed as $\max(\mathcal{H}_a^{(1)}, ..., \mathcal{H}_a^{(N)})$. Finally, we repeat this process for each actor appearing in the scene and accumulate these local scores in the set $\mathcal{S}$. We compute the mean, the maximum, and the minimum of $\mathcal{S}$ and attribute to each frame $f_1, ..., f_N$ the anomaly score (AS) defined as follows:

$$\text{AS}[f_1 : f_N] = \text{mean}(\mathcal{S}) + \log \frac{1 + \max(\mathcal{S})}{1 + \min(\mathcal{S})}. \quad (5)$$

While the $\text{mean}(\mathcal{S})$ summarizes the distribution of the maximum errors of all actors within each frame, the second term takes into account the width of the errors range, as it is mathematically equivalent to:

$$\log\left(1 + \max\left(\mathcal{S}\right)\right) - \log\left(1 + \min\left(\mathcal{S}\right)\right). \quad (6)$$

This increases the anomaly score for spread distributions, which likely correspond to anomalous frames; the logarithm function prevents this term from dominating the final anomaly score.

Algorithm 1 MoCoDAD Train

Require: $X^{1:N}, t, \lambda_1, \lambda_2$
// Divide past from future poses
$\mathcal{P},\ \mathcal{F} = X^{1:k},\ X^{k+1:N}$
// Condition Encoding
$\overline{\mathcal{P}} = \text{E}(\mathcal{P})\ ;\ \ \hat{\mathcal{P}} = \text{D}(\overline{\mathcal{P}})$
$\tau = \tau_\theta(t)$
// Forward Diffusion
$\mathcal{F}_t = q(\mathcal{F}, t)$
// Engender futures
$\hat{\mathcal{F}} = \text{MoCoDAD}(\mathcal{F}_t; \tau, \overline{\mathcal{P}})$
// Loss
$Loss = \lambda_1 \mathcal{L}_{smooth}(\hat{\mathcal{F}}, \mathcal{F}) + \lambda_2 \mathcal{L}_{rec}(\hat{\mathcal{P}}, \mathcal{P})$

D. Further notes on multimodality

We complement Sec. 5.2 of the main paper by showing that multimodality cannot be exploited for separating normal and abnormal classes, since both have a similar degree of diversity (cf. Sec. 3.2).

As for the diversity metrics, we employ the rF metric [3, 12] (see Sec. 3.2) and the *Multimodality* metric proposed in [6].

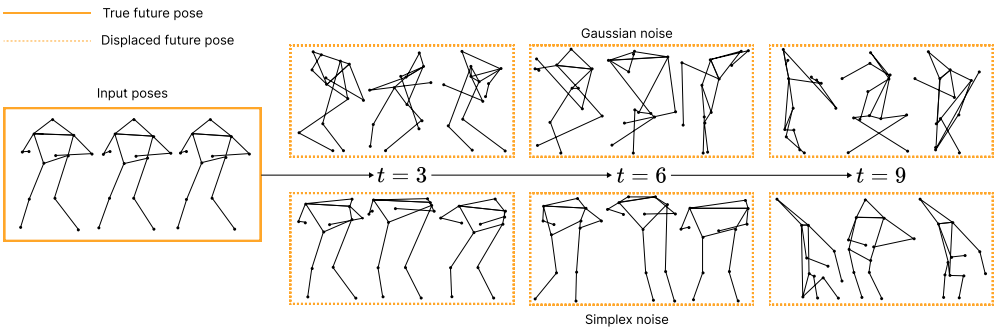

Figure 2: Comparison of Gaussian (up) vs Simplex (down) noises applied to a sequence of future poses.

Algorithm 2 MoCoDAD Inference

Require: $\mathcal{W} = \{f_1, ..., f_N\}$,
$\mathcal{A} = \{\text{actors} \mid \text{actors} \in f_i \; \forall f_i \in \mathcal{W}\}$,
m, T, $\mathcal{G} = \varnothing$, $\mathcal{S} = \varnothing$
for all $a \in \mathcal{A}$ **do**
// Extract and embed past poses
$\mathcal{P}$, $\mathcal{F} = X_a^{1:k}$, $X_a^{k+1:N}$
$\overline{\mathcal{P}} = \text{E}(\mathcal{P})$
// Sample random noise
$\varepsilon \sim \mathcal{N}(0, \mathbf{I})$
// Engender futures
for $j \leftarrow 0$ to m **do**
$\mathcal{F}_{j,T} \leftarrow \varepsilon$
// Reverse diffusion
for $t \leftarrow T$ to 1 **do**
$\tau = \tau_\theta(t)$
$\hat{\mathcal{F}}_j = \text{MoCoDAD}(\mathcal{F}_{j,t}; \tau, \overline{\mathcal{P}})$
// Forward Diffusion
$\mathcal{F}_{j,t-1} = q(\hat{\mathcal{F}}_j, t-1)$
end for
// Get generation anomaly score
$\text{SCORE}_j = \mathcal{L}_{smooth}(\hat{\mathcal{F}}_j, \mathcal{F})$
$\mathcal{G} \leftarrow \mathcal{G} \cup \{\text{SCORE}_j\}$
end for
// Aggregate generations
$\mathcal{H}_a = \text{AGGREGATE}(\mathcal{G})$
$\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{H}_a\}$
end for
// Impute frames' Anomaly Score
$\text{AS}[f_1 : f_N] = \text{mean}(\mathcal{S}) + \log \frac{1+\max(\mathcal{S})}{1+\min(\mathcal{S})}$

Multimodality measures the variance among generated motions given the same conditioning sequence. For each sample s, let $\mathcal{S}$ be the set of all generated motions; then, two subsets $\mathcal{A}(s) = \{\mathbf{a}_1, ..., \mathbf{a}_{S_m}\}$ and $\mathcal{B}(s) = \{\mathbf{b}_1, ..., \mathbf{b}_{S_m}\}$ are sampled from $\mathcal{S}$. Finally, *Multimodality* is given by:

$$\text{Multimodality}(s) = \frac{1}{S_m} \sum_{i=1}^{S_m} \|\mathbf{a}_i - \mathbf{b}_i\|_2 \tag{7}$$

Comparing with Fig. 4 (*right*) of the main paper, the plot in Fig. 1 clearly shows that the anomaly detection performance dramatically drops when assuming a diversity metric as the anomaly score, nearly to random chance. It is worth noting that the performance drops even below random chance when evaluating with rF for a number of generations less than 10.

E. Implementation details

As in [4, 9, 10], we adopt a sliding window procedure for dividing each agent's motion history. We use a window size of 6 frames for all the experiments, of which the first 3 are taken for the condition and the rest for the diffusion process. We adopt similar setups for the imputation proxy tasks (see Sec. 5.4). We set $\lambda_1 = \lambda_2 = 1$. We train the network end-to-end with the Adam optimizer [8] and a learning rate of $1e^{-4}$ with exponential decay for 36 epochs. The diffusion process uses $\beta_1 = 1e^{-4}$ and $\beta_T = 2e^{-2}$, $T = 10$ and the cosine variance scheduler from [11].

Our U-Net-GCN downscales the joints from 17 to 10 and expands the channels from 2 to $(32, 32, 64, 64, 128, 64)$. The conditioning encoder has a channel sequence of $(32, 16, 32)$, with a bottleneck of 32 and a latent projector of 16. We encode the timestep with the positional encoding as defined in [20]. Our training took approximately 7 hours on an Nvidia Quadro P6000 GPU.

F. Results on the UBnormal Validation set

Validation performance is not reported in the main paper, as the validation set is used for hyperparameter fine-tuning. For completeness purposes, as done in [1], we report MoCoDAD's performances vs SoA on the validation set of UBnormal.

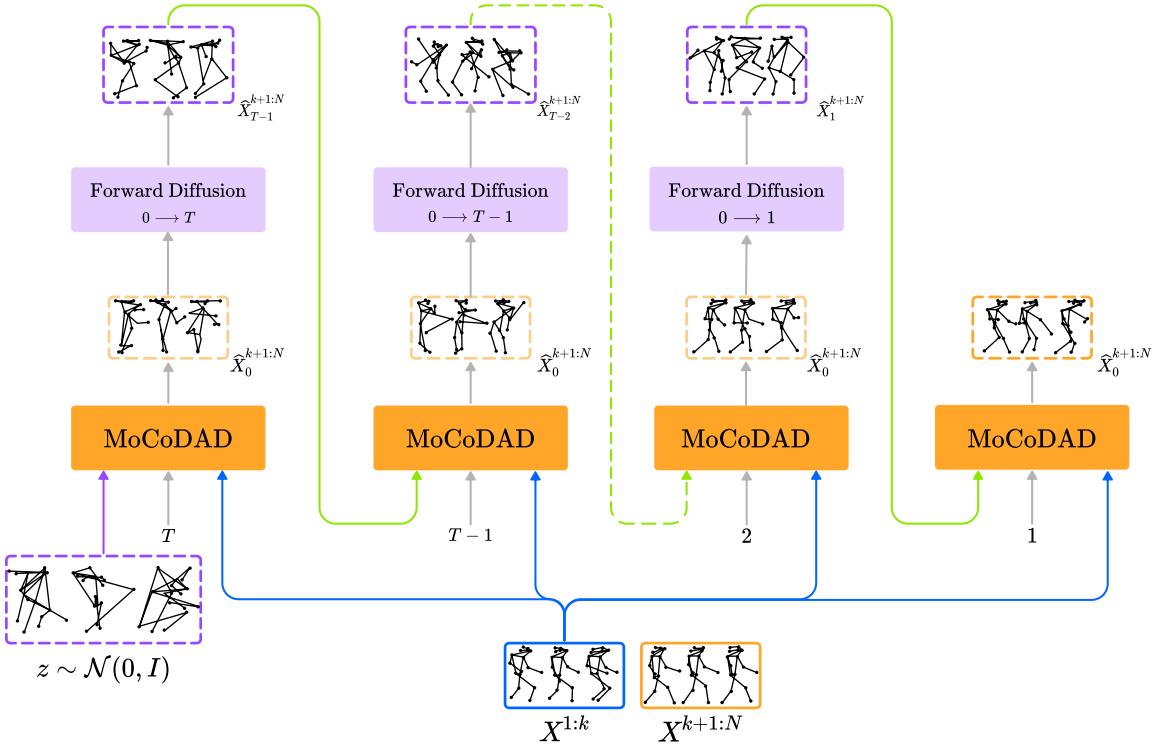

Figure 3: The iterative sampling process of our proposed method (cf. Sec. 3.2 in the main paper). At each step, MoCoDAD generates a prediction (light orange dashed boxes) employing a pose (purple dashed boxes) displaced proportionally to the current timestep t (when $t = T$ we just sample from random noise), together with a prior motion encoding $X^{1:k}$ and the current timestep t. The current prediction is then fed to the Forward Diffusion module, which adds a displacement map to it, anew corrupting the pose proportionally to a smaller timestep. This process is iteratively repeated from T to 1, continuously refining the prediction which is then compared with the actual future (orange box).

Table 3: Comparison of MoCoDAD against SoA in terms of AUC-ROC on the validation set of UBnormal. OCC skeleton-based techniques (∗) are directly comparable to MoCoDAD. Supervised (†) and weakly supervised (‡) methods are also reported, *grayed-out* since they leverage extra annotations.

	UBnormal
Sultani et al. [17] †	51.8
AED-SSMTL [5] †	68.2
TimeSformer [2] †	86.1
AED-SSMTL [5] ‡	58.5
MPED-RNN [10] ∗	61.2
GEPC [9] ∗	47.0
COSKAD [4] ∗	76.4
MoCoDAD ∗	**77.6**

Table 3 shows that the validation set results are in line with those on the test sets reported in Table 1 of the main paper. Notice that MoCoDAD outperforms all the other OCC approaches reaching an AUC of 77.6. Additionally, considering (weakly) supervised approaches that require labeled data (anomalies included), MoCoDAD is only second to TimeSformer [2].

G. Generating motion sequences

This section visually illustrates how a sample is generated using the reverse procedure (Fig. 3). This supplements the discussion presented in Sec. 3 (main paper), providing a visual explanation of Eq. 7. MoCoDAD generates motion sequences depending on a particular conditioning signal, as explained in Section 3 of the main paper. This process is shown graphically in Fig. 3. Random noise x_T in the dimensions corresponding to the desired motion is initially sampled. The process then proceeds iteratively from step T

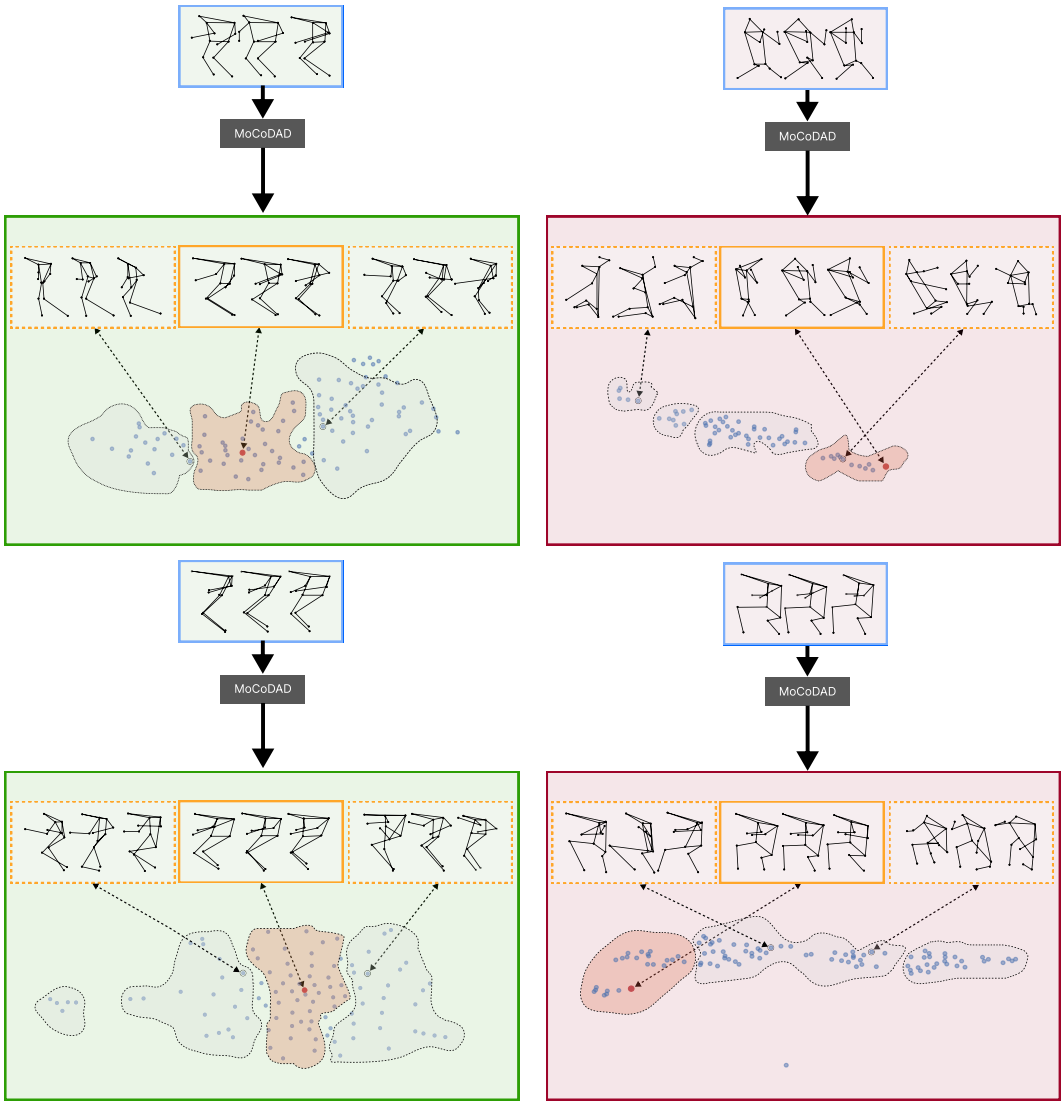

Figure 4: MoCoDAD detects anomalies by synthesizing and statistically aggregating multimodal future motions, conditioned on past frames. Red (right) and green (left) represent examples of anomaly and normality. At the bottom, 100 futures (2d mapped via t-SNE) are generated (dashed-orange rectangles) via a diffusion probabilistic model, conditioned on the past frames (blue-outlined rectangles). Within the distribution modes (highlighted contours), the red dots are the actual true futures corresponding to the sequence of future poses (orange-outlined rectangles). In the case of normality, the true future lies within a main distribution mode, and the generated predictions are pertinent. In the case of abnormality, the true future lies in the tail of the distribution modes, which yields poorer predictions, highlighting anomalies.

to 1. MoCoDAD predicts a clean sample x_0 at each step t, then diffuses back to the previous X_{t-1}.

H. Qualitative results

Fig. 4 reveals that the generations produced with normal conditioning are biased towards the true future. The figure illustrates the t-SNE [19] 2D-embeddings of the generated future frames (orange rectangles), conditioned on the past (blue rectangles). Here, we present two groups of illustrations based on normal (green) and abnormal (red) past, respectively. When the conditioning is normal, the generations (dashed-orange rectangles) are nearby the true motion which lies at the center of the distribution. However, when the past is anomalous, the true future is significantly distant from the center of the distribution produced. Since the diffusion process can generate multiple plausible futures - contoured shapes in the figure - this enforces our assertion that MoCoDAD is multimodal in both normal and anomalous contexts. In the former case, it is capable of generating

samples that are much more pertinent to the actual future; while, in the latter, the generated samples yield poorer predictions, highlighting anomalies (e.g., the first generation in the upper-right corner, and the second generation in the lower-right corner).

References

[1] Andra Acsintoae, Andrei Florescu, Mariana-Iuliana Georgescu, Tudor Mare, Paul Sumedrea, Radu Tudor Ionescu, Fahad Shahbaz Khan, and Mubarak Shah. Ubnormal: New benchmark for supervised open-set video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20143–20153, 2022.

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, page 4, 2021.

[3] Laura Calem, Hedi Ben-Younes, Patrick Pérez, and Nicolas Thome. Diverse probabilistic trajectory forecasting with admissibility constraints. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3478–3484, 2022.

[4] Alessandro Flaborea, Guido D'Amely, Stefano D'Arrigo, Marco Aurelio Sterpa, Alessio Sampieri, and Fabio Galasso. Contracting skeletal kinematics for human-related video anomaly detection, 2023.

[5] 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 (CVPR)*, pages 12742–12752, June 2021.

[6] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[9] Amir Markovitz, Gilad Sharir, Itamar Friedman, Lihi Zelnik-Manor, and Shai Avidan. Graph embedded pose clustering for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10539–10547, 2020.

[10] Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11996–12004, 2019.

[11] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[12] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 282–298. Springer, 2020.

[13] Ken Perlin. Improving noise. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 681–682, 2002.

[14] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

[15] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021.

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.

[17] 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.

[18] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The 11th International Conference on Learning Representations*, 2023.

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

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] Julian Wyatt, Adam Leach, Sebastian M. Schmon, and Chris G. Willcocks. Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 650–656, June 2022.

[22] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Ming-Hsuan Yang, and Bin Cui. Diffusion models: A comprehensive survey of methods and applications. *CoRR*, abs/2209.00796, 2022.