SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Dipartimento di Informatica
PhD in Computer Science

Thesis For The Degree Of Doctor Of Philosophy

Predictive Perception for Detecting Human Motion Anomalies and Procedural Mistakes

Thesis Advisor
Prof. Fabio Galasso

Candidate
Guido Maria D'Amely di Melendugno
1341550

Academic Year MMXX-MMXXIII (XXXVI cycle)

Abstract

Computer Vision emerges as a cornerstone field within Artificial intelligence, enabling digital systems to sense the world through images, mirroring the human ability to see and interpret their surroundings. This ability is paramount, as it allows autonomous systems to interact with humans, promising to reliably extend the applications of AI to productive systems. For example, in Human-Robot collaboration (HRC), accurate vision-based techniques can prevent accidents by providing the cobot with the ability to interpret and swiftly respond to human worker actions. Similarly, in smart manufacturing, Computer Vision methods allow for the timely detection of errors and anomalies in production lines, enhancing quality control and safety, or in video surveillance, where they monitor environments for security threats, promptly identifying unusual behaviors or hazardous situations before they exacerbate. However, the deployment of Computer Vision technologies in real-world scenarios is hampered by significant challenges. These include the requirement for real-time responsiveness, the ability to function reliably in diverse and unpredictable environments, and the development of comprehensive metrics for assessing detection accuracy and system reliability.

This thesis explores machine perception's role in enhancing safety and productive integrity across several domains. By leveraging cutting-edge methodologies such as Denoising Diffusion Probabilistic Models and Large Language Models in novel domains, we propose innovative solutions for applications that require a fine understanding of human behaviors and environments to promote effectiveness, safety, and efficiency.

First, we delve into the HRC domain. Aiming to improve the current methods' efficiency, we devise a lightweight Separable-Sparse Graph Convolutional model that we dub *SeS-GCN*. SeS-GCN bottlenecks the interaction of the GCN's spatial, temporal, and channel-wise dimensions and further learns sparse adjacency matrices by a teacher-student framework. These modeling choices lower the model's memory footprint, providing a practical solution that proves effective both in Human-Pose Forecasting and Collision Avoidance. Moreover, the Cobots and Humans in Industrial COllaboration (CHICO) dataset is proposed to foster research in this field. For the first time, CHICO encompasses 3D-synchronized views and recorded poses of humans and cobots while collaborating in a real industrial scenario, representing a precious resource for advancing safe human-robot collaboration.

Safety often coincides with promptly detecting and responding to mistakes or anomalies, which risk otherwise aggravating, potentially producing dangerous collisions or productive inefficiencies. Thus, following a review of the latest advancements in Video Anomaly Detection methodologies, this thesis builds on the established one-class classification framework, proposing two techniques for human-related Anomaly Detection. The first study investigates adopting non-Euclidean latent spaces to set the one-class-classification's metric objective. We leverage the unique properties of the hyperbolic and spherical metric manifolds for improving human-related anomaly detection. The second proposal introduces a Motion Conditioned Diffusion-based approach for Anomaly Detection (*MoCoDAD*). Indeed, for the first time, MoCoDAD introduces a method for video anomaly detection that exploits cutting-edge diffusive models for spotting anomalies in motion sequences. We review the common reconstruction-based technique, coupling it with the generative ability of diffusion probabilistic models, extending the state-of-the-art in human-related Video Anomaly Detection, and providing relevant insights that serve as the foundation for online mistake detection.

Next, this thesis deals with error anticipation in procedural activities. Acknowledging the absence of a proper benchmark for this task, we apply the insights from the one-class-classification paradigm and Video Anomaly Detection and propose two novel datasets, metrics, and baseline methods for detecting errors in industrial procedural videos. Moreover, we present an innovative technique that exploits the emerging

reasoning capabilities of Large Language Models to detect mistakes in procedural video sequences. This results in a novel multimodal approach that leverages an action recognition module to classify the steps of Egocentric procedural videos and couple it with a Language model to analyze the obtained procedural transcripts and detect mistakes.

This work offers empirical validation through extensive testing on established and newly introduced datasets; bridging the gap between Video Anomaly Detection and Procedural Mistake Detection, it presents a robust foundation for future research and practical applications. We advance the understanding of procedural mistakes as open-set phenomena and emphasize the crucial need for online detection mechanisms, thus enhancing safety and operational efficiency in these environments. These findings lay the foundation for future research, shaping the development of safer, more adaptive industrial automatic systems.

Keywords: Human Motion, Pose Forecasting, Video Anomaly Detection, One-Class Classification, Mistake Detection, Diffusion Probabilistic Models, Procedural Learning, Online Mistake Detection

Contents

List of Figures

List of Tables

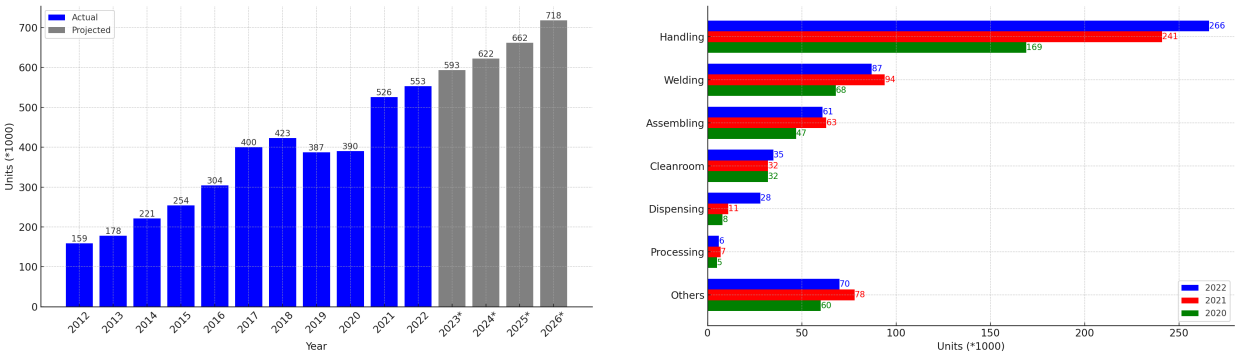
Chapter 1

Introduction

In recent years, a technological revolution has unfolded with the rapid emergence of Artificial Intelligence (AI) and Computer Vision (CV), marking a critical moment for exploring their capabilities and limitations. These technologies have found applications in several different areas, ranging from healthcare [31, 145] and urban infrastructure [18, 76] to entertainment [137, 160] and security [17, 200], reshaping societal functions and offering unprecedented solutions. Particularly notable is the role of these advancements in ushering in the era of Industry 4.0, characterized by increased automation and the integration of smart technologies into manufacturing processes. As reported in the leftmost plot in Fig. 1.1, this evolution is underscored by a significant uptick in the volume of robots installations across industries worldwide, demonstrating a paradigm shift towards more autonomous and efficient production systems. The trend in industrial robot installations is notable, characterized by a substantial 36% increase in 2021 confirmed by a supplementary 5% rise in 2022 relative to previous years. Forecasts suggest a continued upward trajectory, with an anticipated 7% growth rate projected for the ensuing years, surpassing 600000 new units in 2024. Particularly, CV has played a transformative role in industrial domains like manufacturing, logistics, and retail [20, 47, 75, 140]. This focus is further demonstrated by recent trends, as depicted in the rightmost plot in Fig. 1.1, which highlight that the majority of robot installations in the past three years have been directed towards supporting tasks that require high precision and accuracy, such as handling, welding, and assembly. Robots projected for these applications represent 85% of the industrial robot installations in 2022. In particular, handling robots are the most employed in factories, as they can serve to carefully move heavy objects, relieving humans from conducting hazardous work. Indeed, a significant shift has occurred here: workers are moving away from traditional machine operations to more advanced collaborations with robotic systems [21]. In these sectors, the integration of CV has enabled novel, more precise, and efficient interactions between humans and robots. Collaborative robots (cobots), equipped with vision systems can now perform tasks ranging from warehouse inventory sorting to assisting in intricate assembly lines. This synergy of human dexterity and robotic precision is redefining workplace dynamics. This transition heralds exciting innovations, notably accelerating industrial processes and elevating operational efficiency.

However, these advancements bring forth complex challenges, especially in ensuring the safety and reliability of human-robot interactions [21, 85]. Integrating robots into workspaces traditionally occupied by humans presents unique safety challenges. As robots become more prevalent in these environments, the dynamics of human-robot interaction rapidly evolve, requiring stringent safety protocols. A primary concern is thus the physical safety of human workers in proximity to robotic machinery. Unlike traditional automated systems confined to separate spaces, cobots work alongside humans, sharing the same physical space. This

Figure 1.1: Trends of industrial robots installations and their applications over the years. *(left)* Trend of industrial robots installations world-wide. Records of installments from 2012 to 2022 are reported in blue bars; from 2023 to 2026, we report the projections in gray bars. *(right)* Robot installations in 2020-2022 are grouped by their applications. Source: IFR report 2023.



coexistence raises critical questions about preventing accidents and injuries [150, 187]. Robots, designed to operate with high precision and speed, may not inherently possess the awareness or adaptability to respond safely to unpredictable human behavior. The risk of collisions, workers' unintended movements, or mechanical failures could lead to hazardous situations, and the complexity of tasks performed by humans and robots in tandem adds layers of potential safety risks. For instance, a robot might not recognize an unusual or sudden human action, leading to incorrect responses that could compromise safety. Additionally, the industry's goal to expedite production processes involves the adoption of innovative tools such as Egocentric Cameras, Virtual Reality devices, robotic wearables, and advanced monitoring systems. While these tools aim to enhance efficiency, they also introduce increasingly intricate and specialized procedures. A critical question then arises: *how can we ensure error-free workflows without compromising the benefits of these technological advancements?* Critically, addressing this point requires a seamless integration of safety measures that can timely detect anomalies and procedural mistakes and subsequently inform the human operator while maintaining the integrity and efficiency of the enhanced production processes.

This work explicitly addresses these two pivotal challenges: ensuring the safety of human workers in shared spaces with robots and maintaining an error-free workflow in increasingly complex industrial environments. In addressing these challenges, the thesis builds upon recent advances in the CV field, harnessing cutting-edge research to pioneer new solutions. A key innovation is the implementation of an optimized Graph Convolutional Network (GCN), specifically tailored for accurate and swift human pose estimation in industrial settings where humans and cobots collaborate. This approach is instrumental in ensuring safety, as it equips cobots with the capability to rapidly interpret and react to human movements, thereby mitigating the risk of accidents. Furthermore, this thesis advances the exploration of mistake detection in procedural sequences, a task that has recently garnered increasing attention within the Computer Vision community [44, 57, 83, 159, 182]. Despite recent investigations into this area, our analysis reveals a tendency towards oversimplification in existing studies. Notably, prior research has predominantly focused on specific error types, such as omitted or misordered procedural steps, neglecting the inherent unpredictability and open-ended nature of mistakes. Additionally, there is a significant gap in the current methodology, as most existing approaches rely on offline analysis of the procedures, scrutinizing procedures post-completion [44, 57, 159]. This approach contrasts sharply with the practical requirements of real-time feedback in industrial settings, where immediate error correction is crucial for maintaining workflow efficiency and safety. In response, this thesis supports a more nuanced understanding of procedural mistakes as

open-set phenomena and underscores the necessity for online detection mechanisms. This work proposes a novel framework for Online Mistake Detection in procedural videos. This initiative is rooted in an in-depth analysis of the semantically close Video Anomaly Detection (VAD) field, a well-established and sibling domain. By drawing parallels and identifying overlaps between VAD and the emergent field of procedural mistake detection, we aim to sculpt a coherent and practical approach to this new challenge. In this exploration, we leverage and reinterpret two recent milestones in AI: diffusion models [72, 130, 155, 172] and Large Language Models (LLMs) [42, 175]. While these technologies have predominantly contributed to advancements in natural language processing and generative image modeling, their potential in the context of procedural video analysis is untapped. Diffusion models offer a nuanced approach to generating and interpreting complex data patterns, a capability that we adapt for detecting anomalies in human motion and behavior. Similarly, LLMs, known for their emergent deep understanding and reasoning abilities [185], are repurposed in our research to interpret and anticipate procedural actions, providing a unique dimension to mistake detection in an industrial context. Indeed, this research is poised at a crucial juncture where the potential of AI and CV in enhancing industrial processes, particularly in manufacturing, is being rigorously tested and expanded.

The thesis is structured as follows:

In Chapter 2, we focus on Human-Robot Collaboration (HRC). In HRC, Human Pose Forecasting has emerged as a critical area, especially relevant in robotics and autonomous driving. Despite encouraging advancements in terms of performance on established benchmarks [79, 116], we identify and expose two major shortcomings that need to be addressed for the reliable deployment of these systems in real-world applications. Firstly, there are limitations of existing models in terms of size and speed for real-time applications. Such models are not optimized for scenarios where a cobot needs to swiftly and accurately detect the position of a human co-worker to react and thus prevent potential collisions. Secondly, there is a notable lack of a specialized dataset tailored to analyze human-robot interactions in industrial settings, as existing benchmarks [79, 116] primarily focus on general actions such as “walking”, “eating”, or “running”. To address these challenges, our first contribution is the development of the Separable-Sparse Graph Convolutional Network (SeS-GCN). This model builds upon the state-of-the-art STS-GCN [164], designed to be more compact yet equally effective. SeS-GCN innovatively encodes spatial, temporal, and channel-wise dimensions separately within the dynamic pose graph of human key points, achieving state-of-the-art performance and significantly reducing parameters. Additionally, we employ a teacher-student strategy to sparsify the GCN’s adjacency matrix further, reducing computational load and allowing near real-time processing capabilities. Complementing this, we introduce CHICO, a dataset specifically designed for benchmarking human pose forecasting and collision detection in simulated yet realistic industrial scenarios. CHICO includes multi-view videos, 3D poses, and trajectories of human workers and cobots, capturing interactions during specialized tasks like “polishing” or “hammering”. These tasks, typically performed in tandem with cobots, represent the complex dynamics of human-robot collaboration. CHICO thus fills a critical gap in HRC research, providing a robust platform to test and validate models like SeS-GCN in real-world industrial contexts.

The second challenge addressed in this thesis is the under-explored domain of procedural mistake detection in industrial settings. Recognizing the nascent nature of research in this area, we draw a parallel with the well-established VAD field, mainly focusing on human-related anomalies. By starting from the foundational definitions of anomalies and procedural mistakes, we acknowledge both their similarities and differences. This understanding paves the way for adapting and transferring methodologies from VAD to this relatively

uncharted domain, offering a new perspective in the pursuit of effective procedural mistake detection. We delve deeply into the established field of VAD to uncover and leverage the intrinsic connections between anomaly and mistake detection. Our investigation reveals that both domains share foundational elements within the One-Class Classification (OCC) framework, characterized by the rarity and diversity of anomalies and mistakes - making them elusive and challenging to catch. The OCC approach is particularly suited for these tasks as it accommodates a broad spectrum of anomalies, encompassing inherently unprecedented and unknown behaviors. In this context, we introduce two novel methodologies: COSKAD and MoCoDAD.

Chapter 3 presents COSKAD, an innovative model designed for VAD, focusing on the task of detecting anomalies in human behavior. COSKAD leverages the encoding of skeletal human motion through a graph convolutional network, realizing the contraction of skeletal kinematic embeddings onto a latent hypersphere. This technique explores three distinct metric latent spaces — Euclidean, spherical, and hyperbolic — each contributing uniquely to the model's ability to compress normal data embeddings during training into a tightly defined manifold region. Such compression is key to COSKAD's effectiveness, as it enables the model to distinguish anomalies by identifying data points that lie outside the established normality region at inference. In this chapter, we first discuss the One-Class Classification paradigm, which we later transfer into the domain of mistake detection.

Chapter 4 extends this exploration by introducing MoCoDAD. Here, we deepen our analysis of OCC methods, advocating for recognizing the multimodal nature of normal human behaviors, where a single 'normal' action can be performed in various ways. This understanding is crucial to prevent misclassifying unusual yet plausible actions as anomalies. MoCoDAD, for the first time in VAD, combines a Graph Convolutional Network backbone with a Denoising Diffusion Probabilistic Model (DDPM). This integration allows MoCoDAD to generate diverse, plausible future human motions from a given sequence of frames, detecting anomalies based on the deviation of these predictions from the actual future sequences. The greater the divergence, the higher the likelihood of anomalous behavior. Our extensive experiments on established benchmarks demonstrate MoCoDAD's superior predictive accuracy, pushing the boundaries in VAD and offering significant insights into procedural mistake detection.

In Chapter 5 we address the lack of a robust benchmark for Mistake Detection in procedural activities. Our work defines the objectives and constraints essential for an Online Mistake Detection (OMD) framework suitable for industrial applications. We adapt two existing datasets, Assembly101 [159] and Epic-tent [83], for OMD assessment and introduce PREGO, a pioneering method for OMD in egocentric procedural videos. PREGO uniquely combines action recognition and forecasting modules, the latter leveraging the emergent capabilities of Large Language Models for symbolic reasoning. This approach allows for adaptable, procedure-agnostic Mistake Detection, significantly improving procedural accuracy and safety in industrial environments. we highlight the insights of this work and elaborate on future studies, respectively. To summarize, the contribution of this thesis is threefold:

- We advance real-time safety and operational efficiency in industrial settings by presenting a novel method specifically tailored for HRC and defining a novel framework for Online Mistake Detection. This innovation enhances human-robot collaboration and procedural integrity, enabling immediate detection and correction of procedural mistakes.

- We harness cutting-edge AI technologies such as diffusion models and LLMs in novel visual contexts. Our work demonstrates their utility in detecting procedural mistakes and anomalies in real-time, broadening their application in visual data analysis and contributing to the evolution of intelligent in-

dustrial systems.

- We provide empirical validation and theoretical contributions through extensive testing on established and newly introduced datasets. Our comprehensive analysis bridges the gap between Video Anomaly Detection and procedural mistake detection, offering a robust foundation for future research and practical applications in the field.

This transformative shift in industrial practices, fueled by advancements in AI and Computer Vision, represents a strategic step forward, grounded in the extensive research and development efforts that have characterized the field for years. It's a challenge that CV has been evolving towards, preparing to address with a wealth of research and innovation. Acknowledging this, it becomes imperative to analyze and build upon these established domains and methods. By doing so, we can develop informed strategies and advanced systems tailored to meet the emerging challenges of this integration. This thesis aims to contribute to this critical transition by refining and applying proven CV techniques in new, dynamic industrial contexts. The goal is not just to adapt to this change but to lead it, ensuring that the shift towards more automated and intelligent systems is both seamless and robust, grounded in years of research and innovation.

Chapter 2

Pose Forecasting in Industrial Human-Robot Collaboration

2.1 Overview

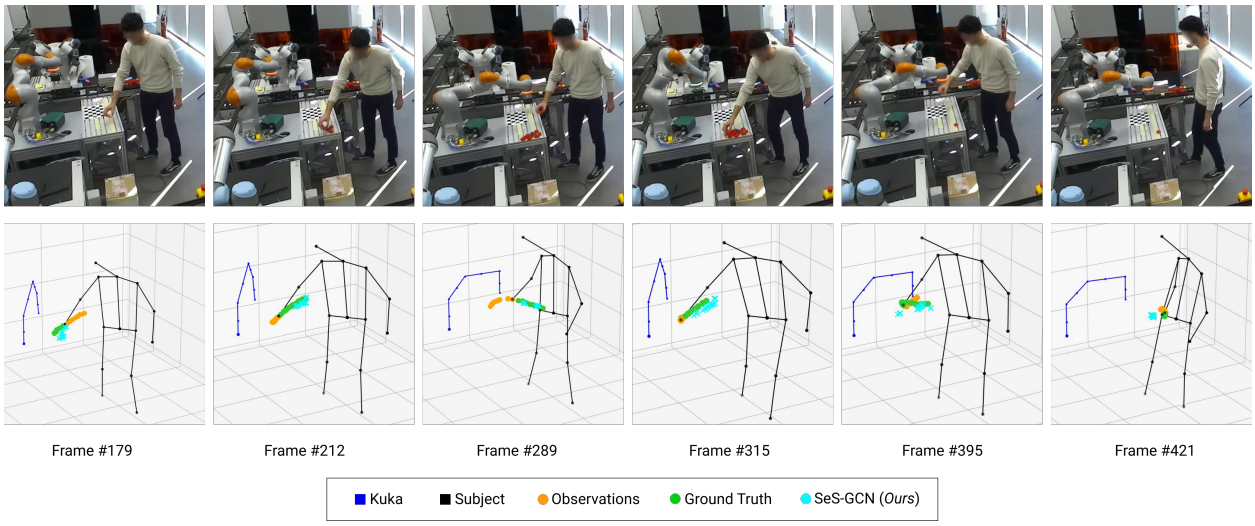


Figure 2.1: A collision example from our CHICO dataset. On the top row some frames of the *Lightweight pick and place* action captured by one of the three cameras. On the bottom row, operator + robot skeletons. The forecasting takes an observation sequence (in yellow, here pictured for the right wrist only), and performs a prediction (cyan) which is compared with the ground truth (green). On frame 395 is it easy to see the robot hitting the operator, which is retracting, as is evident in frame 421. See how the predictions by SeS-GCN follow closely the GT, except during the collision. At collision time, due to the impact, the abrupt change of the arm motion produces uncertain predictions, as it shows by the very irregular predicted trajectory.

Collaborative robots (cobots) and modern Human-Robot Collaboration (HRC) depart from the traditional separation of functions of industrial robots [93] because of the shared workspace [80]. Additionally, cobots and humans perform actions concurrently, and they will, therefore, physically engage in contact. While there is a clear advantage in increased productivity [152] (improved by as much as 85% [162]) due to the minimization of idle times, there are challenges in the workplace safety [62]: it is not about whether there will be contact, but rather about understanding its consequences [121].

The pioneering work of Shah et al. [162] has already shown that, in order to seamlessly and efficiently interact with human co-workers, cobots need to abide by two collaborative principles: (1) Making decisions

on-the-fly, and (2) Considering the consequences of their decision on their teammates. The first calls for promptly and accurately detecting human motion in the workspace. The second principle implies that cobots need to anticipate the pose trajectories of their human co-workers and predict future collisions.

Motivated by these problems, the first contribution of our work is a novel Separable-Sparse Graph Convolutional Neural Network (SeS-GCN) for human pose forecasting. Pose forecasting requires an understanding of the complex spatio-temporal joint dynamics of the human body, and recent trends have highlighted the promises of modeling body kinematics within a single GCN framework [37, 39, 101, 104, 118, 181, 202]. We have designed SeS-GCN with performance and efficiency in mind by bringing together, for the first time, three main modeling principles: depthwise-separable graph convolutions [95], space-time separable graph adjacency matrices [164], and sparse graph adjacency matrices [163]. In SeS-GCN, *separable* stands for limiting the interplay of joints with others (space), at different frames (time) and per channel (depth-wise). Within the GCN, different channels, frames, and joints still interact by means of multi-hop messages. For the first time, sparsity is achieved by a teacher-student framework. The reduced interaction and sparsity results in fewer parameters than all GCN-based baselines [95, 163, 164] - $\sim$75% fewer parameters with respect to a fully learnable GCN, see Tab. 2.2- while improving performance by more than 2.7%. Compared to the state-of-the-art (SoA) [117], SeS-GCN is lightweight, only using 1.72% of its parameters, it is $\sim$4 times faster while remaining competitive with just 1.5% larger error on Human 3.6M [79] when predicting 1 sec in the future.

The model is described in detail in Sec. 2.3, and experiments and ablation studies are illustrated in Sec. 2.5.

We also introduce the very first benchmark of Cobots and Humans in Industrial COllaboration (CHICO, an excerpt in Fig. 2.1). CHICO includes multi-view videos, 3D poses and trajectories of the joints of 20 human operators in close collaboration with a robotic arm *KUKA LBR iiwa* within a shared workspace. The dataset features 7 realistic industrial actions taken at a real industrial assembly line with a marker-less setup. The goal of CHICO is to endow cobots with perceptive awareness to enable human-cobot collaboration with contact. Towards this frontier, CHICO proposes benchmarking two key tasks: human pose forecasting and collision detection. Cobots currently detect collisions by mechanical-only events (transmission of contact wrenches, control torques, sensitive skins). This ensures safety, but it harms the human-cobot interaction because collisions break the motion of both, which reduces productivity and may be annoying to the human operator.

CHICO features 240 1-minute video recordings, from which two separate sets of test sequences are selected: one for estimating the accuracy in pose forecasting, so cobots may be aware of the next future (1.0 sec); and one with 226 genuine collisions, so cobots may foresee them and possibly re-plan. The dataset is detailed in Sec. 2.4, and experiments are illustrated in Sec. 2.6.

When tested on CHICO, the proposed SeS-GCN outperforms all baselines and reaches an error of 85.3 mm (MPJPE) at 1.00 sec, with a negligible run time of 2.3 msec (as reported in Table 2.5). Additionally, the forecast human motion is used to detect human-cobot collision by checking whether the predicted trajectory of the human body intersects that of the cobot. This is also encouraging, as SeS-GCN allows to reach an F1-score of 0.64. Both aspects contribute to a cobot awareness of the future, which is instrumental for HRC in industrial applications.

Table 2.1: Comparison between the state-of-the-art datasets and the proposed CHICO; *unk* stands for "unknown".

	Quantitative Details							Rec. Scene	Actions Type		Tasks			Markerless
	# Classes	# Subj.	Avg Rec. Time	# Joints	FPS	Aspect Ratio	# Sensors		Industr.	HRC	Action Recog.	Pose Forec.	Coll. Det.	
Human3.6M [79]	15	11	100.49 s	32	25	normalized	15	mo-cap studio				✓		
AMASS [116]	11265	344	12.89 s	variable	variable	original	variable	mo-cap studio				✓		
3DPW [180]	47	7	28.33 s	18	60	original	18	outdoor locations				✓		
ExPI [63]	16	4	*unk*	18	25	original	88	mo-cap studio				✓		
CHI3D [50]	8	6	*unk*	*unk*	*unk*	original	14	mo-cap studio				✓		
InHARD [38]	14	16	< 8 s	17	120	original	20	assembly line	✓	✓	✓			
CHICO (ours)	7	20	55 s	15	25	original	3	assembly line	✓	✓		✓	✓	✓

2.2 Related Work

Human pose forecasting. Human pose forecasting is a recent field which has some intersection with human action anticipation in computer vision [101] and HRC [45]. Previous studies exploited Temporal Convolutional Networks (TCNs) [7, 54, 100, 136] and Recurrent Neural Networks (RNNs) [52, 60, 81]. Both architectures are naturally suited to model the temporal dimension. Recent works have expanded the range of available methods by using Variational Auto-Encoders [23], specific and model-agnostic layers that implicitly model the spatial structure of the human skeleton [4], or Transformer Networks [24].

Pose forecasting using Graph Convolutional Networks (GCN). Most recent research uses GCNs [39, 104, 117, 164, 202]. In [117], the authors have mixed GCN for modelling the joint-joint interaction with Transformer Networks for the temporal patterns. Others [104, 164, 202] have adopted GCNs to model the space-time body kinematics, devising, in the case of [39], hierarchical architectures to model coarse-to-fine body dynamics.

We identify three main research directions for improving efficiency in GCNs: **i.** space-time separable GCNs [164], which factorizes the spatial joint-joint and temporal patterns of the adjacency matrix; **ii.** depth-wise separable graph convolutions [74], which has been explored by [8] in the spectral domain; and **iii.** sparse GCNs [163], which iteratively prunes the terms of the adjacency matrix of a GCN. Notably, all three techniques also yield better performance than the plain GCN. Here, for the first, we bring together these three aspects into an end-to-end space-time-depthwise-separable and sparse GCN. The three techniques are complementary to improve both efficiency and performance, but their integration requires some structural changes (*e.g.*, adopting teacher-student architectures for sparsifying), as we describe in Sec. 2.3.

Human Robot Collaboration (HRC). HRC is the study of collaborative processes where human and robot agents work together to achieve shared goals [11, 28]. Computer vision studies on HRC are mostly related to pose estimation [27, 53, 98] to locate the articulated human body in the scene.

In [32, 86, 128], methodologies for robot motion planning and collision avoidance are proposed; their study perspective is opposite to ours, since we focus on the human operator. In this regard, the works of [12, 36, 87, 106] model the operators' whereabouts through detection algorithms which approximate human shapes using simple bounding boxes. Approaches that predict the human motion during collaborative tasks are in [174, 199] using RNNs and in [179] using Guassian processes. Other work [96] models the upper body and the human right hand (which they call the Human End Effector) by considering the robot-human handover phase. As motion prediction engine, DCT-RNN-GCN [117] is considered, against which we compare in the experiments.

Datasets for pose forecasting. Human pose forecasting datasets cover a wide spectrum of scenarios, see Table 2.1 for a comparative analysis. Human3.6M [79] considers everyday actions such as conversing, eating, greeting and smoking. Data were acquired using a 3D marker-based motion capture system, composed of 10 high-speed infrared cameras. AMASS [116] is a collection of 15 datasets where daily actions were captured by an optical marker-based motion capture. Human3.6M and AMASS are standard benchmarks for human pose forecasting, with some overlap in the type of actions they deal with. The 3DPW dataset [180] focuses on outdoor actions, captured with a moving camera and 17 Inertial Measurement Units (IMU), embedded on a special suit for motion capturing [148]. The recent ExPI dataset [63] contains 16 different dance actions performed by professional dancers, for a total of 115 sequences, and it is aimed at motion prediction. ExPI has been acquired with 68 synchronised and calibrated color cameras and a motion capture system with 20 mocap cameras. Finally, the CHI3D dataset [50] reports 3D data taken from MOCAP systems to study human interactions.

None of these datasets answer our research needs, i.e., a benchmark taken by a sparing, energy-efficient markerless system, focused on the industrial HRC scenario, where forecasting may be really useful for anticipating collisions between the humans and robots. In fact, the only dataset relating to industrial applications is InHARD [38]. Therein, humans are asked to perform an assembly task while wearing inertial sensors on each limb. The dataset is designed for human action recognition, and it involves 16 individuals performing 13 different actions each, for a total of 4800 action samples over more than 2 million frames. Despite showcasing a collaborative robot, in this dataset the robot is mostly static, making it unsuitable for collision forecasting.

2.3 Methodology

We build an accurate, memory efficient and fast GCN by bridging three diverse research directions: **i.** Space-time separable adjacency matrices; **ii.** Depth-wise separable graph convolutions; **iii.** Sparse adjacency matrices. This results in an all-Separable and Sparse GCN encoder for the human body kinematics, which we dub SeS-GCN, from which the future frames are forecast by a Temporal Convolutional Network (TCN).

2.3.1 Background

Problem Formalization. Pose forecasting is formulated as observing the 3D coordinates $\boldsymbol{x}_{v,t}$ of V joints across T frames and predicting their location in the K future frames. For convenience of notation, we gather the coordinates from all joints at frame t into the matrix $X_t = [\boldsymbol{x}_{v,t}]_{v=1}^{V} \in \mathbb{R}^{3\times V}$. Then we define the tensors $\mathcal{X}_{in} = [X_1, X_2..., X_T]$ and $\mathcal{X}_{out} = [X_{T+1}, X_{T+2}..., X_{T+K}]$ that contain all observed input and target frames, respectively.

We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to encode the body kinematics, with all joints at all observed frames as the node set $\mathcal{V} = \{\boldsymbol{v}_{i,t}\}_{i=1,t=1}^{V,T}$, and edges $(\boldsymbol{v}_{i,t}, \boldsymbol{v}_{j,s}) \in \mathcal{E}$ that connect joints i, j at frames t, s.
Graph Convolutional Networks (GCN). A GCN is a layered architecture:

$$\mathcal{X}^{(l+1)} = \sigma\left(A^{(l)}\mathcal{X}^{(l)}W^{(l)}\right) \tag{2.1}$$

The input to a GCN layer l is the tensor $\mathcal{X}^{(l)} \in \mathbb{R}^{C^{(l)}\times V\times T}$ which maintains the correspondence to the V body joints and the T observed frames, but increases the depth of features to $C^{(l)}$ channels. $\mathcal{X}^{(1)} = \mathcal{X}_{in}$ is the input tensor at the first layer, with $C^{(1)} = 3$ channels given by the 3D coordinates. $A^{(l)} \in \mathbb{R}^{VT\times VT}$ is

the adjacency matrix relating pairs of VT joints from all frames. Following most recent literature [39, 117, 163, 164], $A^{(l)}$ is learnt. $W^{(l)} \in \mathbb{R}^{C^{(l)} \times 1 \times 1}$ are the learnable weights of the graph convolutions. σ is a the non-linear PReLU activation function.

2.3.2 Separable & Sparse Graph Convolutional Networks (SeS-GCN)

We build SeS-GCN by integrating the three mentioned modelling dimensions: **i.** separating spatial and temporal interaction terms in the adjacency matrix of a GCN; **ii.** separating the graph convolutions depth-wise; **iii.** sparsifying the adjacency matrices of the GCN.

Separating space-time. STS-GCN [164] has factored the adjacency matrix $A^{(l)}$ of the GCN, at each layer l, into the product of two terms $A_s^{(l)} \in \mathbb{R}^{V \times V \times T}$ and $A_t^{(l)} \in \mathbb{R}^{T \times T \times V}$, respectively responsible for the temporal-temporal and joint-joint relations. The GCN formulation becomes:

$$\mathcal{X}^{(l+1)} = \sigma\left(A_s^{(l)} A_t^{(l)} \mathcal{X}^{(l)} W^{(l)}\right) \tag{2.2}$$

Eq. (2.2) bottlenecks the interplay of joints across different frames, implicitly placing more emphasis on the interaction of joints on the same frame ($A_s^{(l)}$) and on the temporal pattern of each joint ($A_t^{(l)}$). This reduces the memory-footprint of a GCN by approx. 4x while improving its performance (cf. Sec. 2.5.1). Note that this differs from alternating spatial and temporal modules, as it is done in [196] and [15], respectively for trajectory forecasting and action recognition.

Separating depth-wise. Inspired by depth-wise convolutions [34, 74], the approach in [95] has introduced depth-wise graph convolutions for image classification, followed by [8] which resorted to a spectral formulation of depth-wise graph convolutions for graph classification. Here we consider depth-wise graph convolutions for pose forecasting. The depth-wise formulation bottlenecks the interplay of space and time (operated by the adjacency matrix $A^{(l)}$) with the channels of the graph convolution $W^{(l)}$. The resulting all-separable model which we dub STS-DW-GCN is formulated as such:

$$\mathcal{H}^{(l)} = \gamma\left(A_s^{(l)} A_t^{(l)} \mathcal{X}^{(l)} W_{DW}^{(l)}\right) \tag{2.3a}$$

$$\mathcal{X}^{(l+1)} = \sigma\left(\mathcal{H}^{(l)} W_{\text{MLP}}^{(l)}\right) \tag{2.3b}$$

Adding the depth-wise graph convolution splits the GCN of layer l into two terms. The first, Eq. (2.3a), focuses on space-time interaction and limits the channel cross-talk by the use of $W_{DW}^{(l)} \in \mathbb{R}^{\frac{C^{(l)}}{\alpha} \times 1 \times 1}$, with $1 \leq \alpha \leq C^{(l)}$ setting the number of convolutional groups ($\alpha = C^{(l)}$ is the plain single-group depth-wise convolution). The second, Eq. (2.3b), models the intra-channel communication just. This may be understood as a plain (MLP) 1D-convolution with $W_{\text{MLP}}^{(l)} \in \mathbb{R}^{C^{(l)} \times 1 \times 1}$ which re-maps features from $C^{(l)}$ to $C^{(l+1)}$. γ is the ReLU6 non-linear activation function. Overall, this does not significantly reduce the number of parameters, but it deepens the GCN without over-smoothing [132], which improves performance (see Sec. 2.5.1 for details).

Sparsifying the GCN. Sparsification has been used to improve the efficiency (memory and, in some cases, runtime) of neural networks since the seminal pruning work of [97]. [163] has sparsified GCNs for trajectory forecasting. This consists in learning masks $\mathcal{M}$ which selectively erase certain parameters in the adjacency matrix of the GCN. Here we integrate sparsification with the all-separable GCN design, which yields our

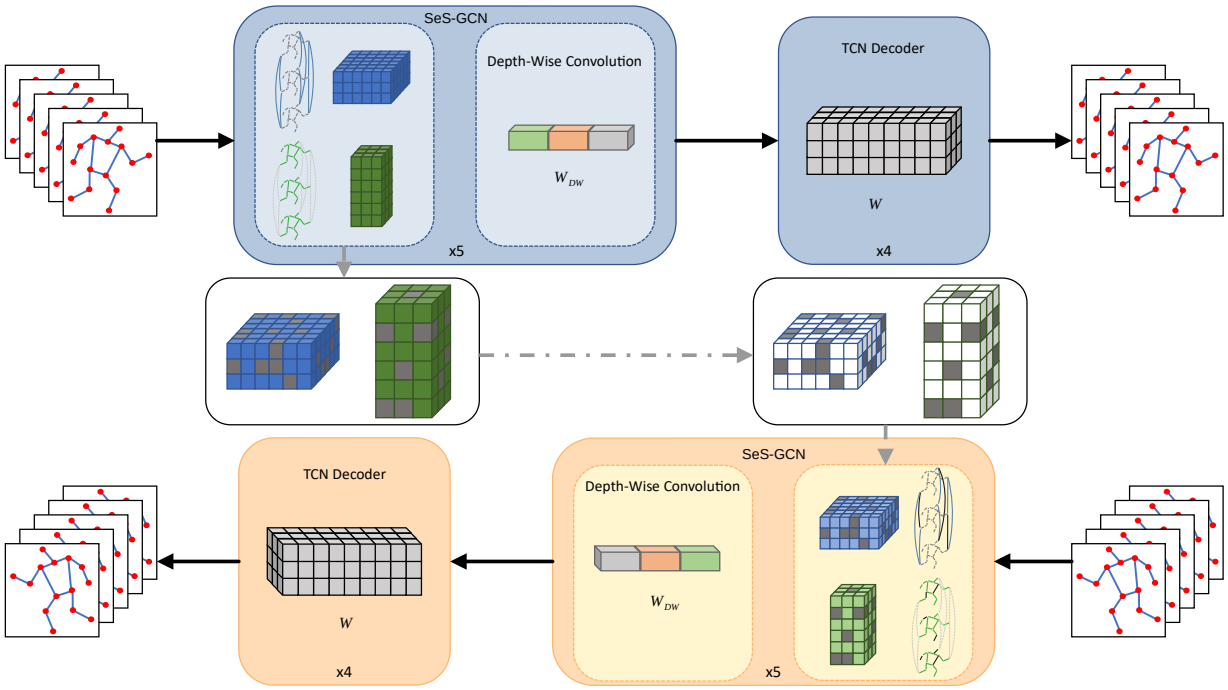


Figure 2.2: Overview of the proposed pipeline. Given a sequence of observed 3D poses, the Teacher network (depicted with blue boxes) encodes the spatio-temporal body dynamics with 5 SeS-GCN layers, composed by the space-time separable encoder followed by the Depth-Wise convolution. The future trajectories are then predicted with 4 TCN layers. After the training of the Teacher, we threshold the values of the spatial and temporal adjacency matrix to obtain the masks, which are then applied during the Student model (depicted in orange boxes) training.

proposed SeS-GCN for human pose forecasting:

$$\mathcal{H}^{(l)} = \gamma\left((\mathcal{M}_s^{(l)} \odot A_s^{(l)})(\mathcal{M}_t^{(l)} \odot A_t^{(l)})\mathcal{X}^{(l)}W_{DW}^{(l)}\right) \tag{2.4a}$$

$$\mathcal{X}^{(l+1)} = \sigma\left(\mathcal{H}^{(l)}W_{\text{MLP}}^{(l)}\right) \tag{2.4b}$$

$\odot$ is the element-wise product and $\mathcal{M}_{\{s,t\}}^{(l)}$ are binary masks. Both at training and inference, [163] generates masks, it uses those to zero certain coefficients of the adjacency matrix A, and it adopts the resulting GCN for trajectory forecasting. By contrast, we adopt a teacher-student framework during training. The teacher learns the masks, and the student only considers the spared coefficients in A. At inference, our proposed SeS-GCN only consists of the student, which simply adopts the learnt sparse A_s and A_t. Compared to [163], the approach of SeS-GCN is more robust at training, it yields fewer model parameters at inference ($\sim$30% less for both A_s and A_t), and it reaches a better performance, as it is detailed in Sec. 2.5.1.

2.3.3 Decoder Forecasting

Given the space-time representation, as encoded by the SeS-GCN, the future frames are then decoded by using a temporal convolutional network (TCN) [7, 54, 100, 164]. The TCN remaps the temporal dimension to match the sought output number of predicted frames. This part of the model is not considered for improvement because it is already efficient and it performs satisfactorily.

2.3.4 Implementation details

The proposed SeS-GCN is written in Pytorch. The model adopts residual connections at each GCN layer, it is regularized with batch normalization [77] at the end of each GCN layer, and it is optimized with ADAM [90]. On Human3.6M [79], training of SeS-GCN proceeds for 60 epochs for both the teacher and the student models. We used batch size of 256, learning rate of 0.1, and decay rate of 0.1 at epochs 5, 20, 30 and 37. On an Nvidia RTX 2060 GPU, the learning process takes 30 minutes.

2.4 The CHICO dataset

This section details the CHICO dataset by describing the acquisition scenario and devices, the cobot and the performed actions. We release RGB videos, skeletons and calibration parameters *.

The scenario. We are in a smart-factory environment, with a single human operator standing in front of a $0.9\,\mathrm{m} \times 0.6\,\mathrm{m}$ workbench and a cobot at its end (see Fig.2.1). The human operator has some free space to turn towards some equipment and carry out certain assembly, loading and unloading actions [122]. In particular, light plastic pieces and heavy tiles, a hammer, and abrasive sponges are available. This table setup is designed with flexibility and practicality, making it generalizable to various industrial scenarios. The compact size of the workbench is suitable for environments with space constraints, and its ergonomic design minimizes operator movement, enhancing productivity and reducing fatigue. The availability of diverse materials and tools illustrates the setup's versatility in handling different assembly tasks, from delicate to more robust operations. The detailed setups for each action are reported graphically in Figures 2.3 -2.9. A total of 20 human operators have been hired for this study. They attended a course on how to operate with the cobot and signed an informed consent form prior to the recordings.

The collaborative robot. A 7 degrees-of-freedom Kuka LBR iiwa 14 R820 collaborates with the human operator during the data acquisition process. Weighing in at $29.5\,\mathrm{kg}$ and with the ability to handle a payload up to $14\,\mathrm{kg}$, it is widely used in modern production lines. More details on the cobot can be found in Sec. 2.4.1.

The acquisition setup. The acquisition system is based on three RGB HD cameras providing three different viewpoints of the same workplace: two frontal-lateral and one rear view. The frame rate is 25Hz. The videos were first checked for erroneous or spurious frames; then, we used Voxelpose [176] to extract a 3D human pose for each frame. Extrinsic parameters of each camera are estimated w.r.t. the robot's reference frame by means of a calibration chessboard of 1×1m, and temporal alignment is guaranteed by synchronization of all the components with an Internet Time Server. In our environment, Voxelpose estimates a joint positioning accuracy in terms of Mean Per Joint Position Error (MPJPE) of 24.99mm using three cameras, which is enough for our purposes as an ideal compromise between the system's portability and accuracy. We confirm these numbers in two ways: the first is by checking that human-cobot collisions were detected with 100% F1 score (we have a collision when the minimum distance between the human limbs and the robotic links is below a predefined threshold). Secondly, we show that the new CHICO dataset does not suffer from a trivial zero velocity solution [120], *i.e.* results achieved by a zero velocity model underperform the current SoA in equal proportion as for the large-scale established Human3.6M.

Actions. The 7 types of actions of CHICO are inspired by ordinary work sessions in an HRC disassembly line as described in the review work of [71]. Each action is repeated over a time interval of $\sim$1 minute on

*Code and dataset are available at: `https://github.com/AlessioSam/CHICO-PoseForecasting`.

average. Each action is associated with a goal that the human operator has to achieve by a given time limit, which requires them to move with a certain velocity. Each action consists of repeated interactions with the robot (*e.g.*, robot place, human picks) which, due to the limited space, lead to some *unconstrained collisions*[*] which we label accordingly. Globally, from the 7 actions $\times$ 20 operators, we collect 226 different collisions. The selected actions are designed to be generalizable to real industrial scenarios, reflecting typical tasks found in various manufacturing and assembly environments. On the next page, an illustration of the actions is available. Note that in all the captions "***H***" stands for "human operator", "***R***" for "robot". In the following, each action is briefly described.

- ***Lightweight pick and place*** (*Light P&P*). The human operator is required to move small objects of approximately 50 grams from a loading bay to a delivery location within a given time slot. The bay and the delivery location are at the opposite sides of the workbench. Meanwhile, the robot loads on of this bay so that the human operator has to pass close to the robotic arm. In many cases, the distance between the limbs and the robotic arm is a few centimeters.

- ***Heavyweight pick and place*** (*Heavy P&P*). The setup of this action is the same as before, but the objects to be moved are floor tiles weighing $0.75\,\mathrm{kg}$. This means that the actions have to be carried out with two hands.

- ***Surface polishing*** (*Polishing*). This action was inspired by [115], where the human operator polishes the border of a 40 by 60cm tile with some abrasive sponge, and the robot mimics a visual quality inspection.

- ***Precision pick and place*** (*Prec. P&P*). The robot places four plastic pieces in the four corners of a 30$\times$30cm table in the center of the workbench, and the human has to remove them and put them on a bay before the robot repeats the same unloading.

- ***Random pick and place*** (*Rnd. P&P*). Same as the previous action, except for the plastic pieces, which are continuously placed by the robot randomly on the central 30$\times$30cm table, and the human operator has to remove them.

- ***High shelf lifting*** (*High lift*). The goal is to pick light plastic pieces (50 grams each) on a sideway bay filled by the robot, putting them on a shelf located at 1.70m, on the opposite side of the workbench. Due to the geometry of the workspace, the arms of the human operator were required to pass above or below the moving robotic arm. In this way, close distances between the human arm and forearm and the robotic links were realized.

- ***Hammering*** (*Hammer*). The operator hits a metallic tide with a hammer held by the robot. In this case, the interest was to check how much the collision detection is robust to an action where the human arm is colliding close to the robotic arm (that is, on the metallic tile) without properly colliding *with the robotic arm*.

Overall, the selected actions represent a broad spectrum of industrial tasks, emphasizing the setup's applicability and realism in simulating effective human-robot collaboration across various real-world scenarios.

[*]Unconstrained collisions is a term coming from [66], indicating a situation in which only the robot and human are directly involved in the collision.

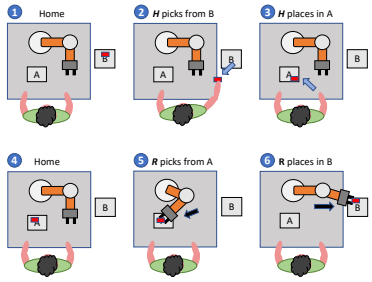


Figure 2.3: ***Light P&P***. A single item (the red brick) is shown here for clarity. In practice, a dozen items were available.

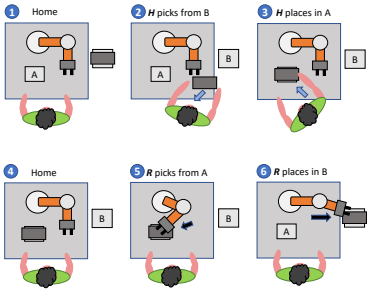


Figure 2.4: ***Heavy P&P***. Moving the object with two hands requires rotating the torso, which partially hides the robot from the operator.

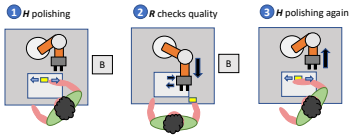


Figure 2.5: ***Polish***. The human has an abrasive sponge used to remove some material from the metallic tile. This action requires the user to be prone on the surface to polish, blocking the robot's view.

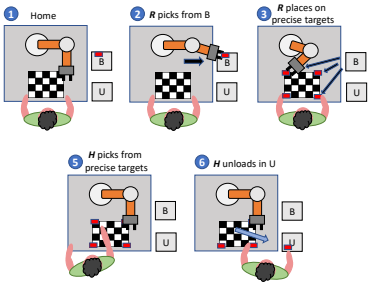


Figure 2.6: ***Prec. P&P***.This action allows us to measure how precise the prediction is in individuating endpoints that will be targeted by the human operator.

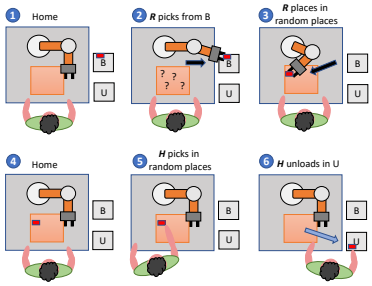


Figure 2.7: ***Rnd. P&P***. The robot puts objects randomly in the workplace, creating collisions during the interaction. A single item (the red brick) is shown here for clarity.

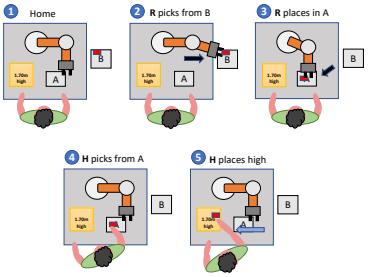


Figure 2.8: ***High Shelf***. The action requires lifting some plastic objects. The operator moves very close to the cobot during this action.

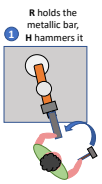


Figure 2.9: ***Hammer***. This action requires the human to be very close to the robot, keeping the item hammer with one hand and the other doing the hammering action.

Table 2.2: MPJPE error (millimeters) for long-range predictions (25 frames) on Human3.6M [79] and numbers of parameters. Best figures overall are reported in bold, while underlined figures represent the best in each block. The proposed model has comparable or less parameters than the GCN-based baselines [74, 163, 164] and it outperforms the best of them [164] by 2.6%.

	Depth	MPJPE	Parameters (K)	DW-Separable	ST-Separable	Sparse	w/ MLP layers	Teacher-Student
GCN	4	123.2	222.7					
DW-GCN [95]	4+4	119.8	223.2	✓			✓	
STS-GCN† [164]	4	<u>117.0</u>	**<u>57.6</u>**		✓			
Sparse-GCN [163]	4	122.7	257.9			✓		
STS-GCN	5	115.9	<u>68.6</u>		✓			
STS-GCN	6	116.1	79.9		✓			
STS-GCN w/ MLP	5+5	125.2	101.4		✓		✓	
STS-DW-GCN	5+5	<u>114.8</u>	70.0	✓	✓		✓	
STS-DW-Sparse-GCN	5+5	115.7	122.4	✓	✓	✓	✓	
SeS-GCN (proposed)	5+5	**<u>113.9</u>**	<u>58.6</u>	✓	✓	✓	✓	✓

2.4.1 Details on the dataset and the data acquisition process

The dataset has been acquired in the October '21 - March '22 period, on a $500m^2$ Industry 4.0 lab, which includes a configurable a 11m production line, 4 cobots, a quality inspection cell, a (dis-)assembly station and other equipment. We worked on the 0.9 m ×0.6 m workbench of the (dis-)assembly station in front of a Kuka LBR iiwa 14 R820 cobot. The declared positioning accuracy is $\pm 0.1\,\mathrm{mm}$ and the axis-specific torque accuracy is $\pm 2\,\%$ [94]. Thanks to its joint torque sensors, the robot can detect contact and reduce its level of force and speed, being compliant to the ISO/TS 15066:2016 [80] standard. Since collisions between the operator and the cobot were expected in CHICO, the maximum allowed Cartesian speed of each link is set to $200\,\mathrm{mm/s}$, slightly lower than the ISO/TS 15066:2016 requirements. The safety torque limit allowed before the mechanical brakes activation is set to $30\,\mathrm{N\,m}$ for all joints and $50\,\mathrm{N\,m}$ for the end-effector. Additionally, a programmable safety check of $10\,\mathrm{N}$ was set on the Cartesian force.

A total of 20 subjects (14 males, 6 females, and average age of 23 years) have been hired for building the dataset. They worked for the entire acquisition period after having signed informed consent and participated in a crash course on how to cooperate with the Kuka cobot. During the acquisition season, we have selected some excerpts that capture collisions made by the operators, reaching 226 collisions. On average, we have 45 collisions for each action, with the sole exception for *hammering*. For this specific action, the cobot stands still, holding the object being hammered, while the human agent moves repeatedly close to the robotic arm. Sequences containing this action are still part of the collision detection dataset, i.e., they are useful to check that there are no false positives.

2.5 Experiments on Human3.6M

We benchmark the proposed SeS-GCN model on the large and established Human3.6M [79]. In Sec. 2.5.1, we analyze the design choices corresponding to the models discussed in Sec. 2.3, then we compare with the state-of-the-art in Sec. 2.5.2.

Human3.6M [79] is an established dataset for pose forecasting, consisting of 15 daily life actions (e.g. Walking, Eating, Sitting Down). From the original skeleton of 32 joints, 22 are sampled as the task, representing the body kinematics. A total of 3.6 million poses are captured at 25 fps. In line with the literature [39, 117, 120], subjects 1, 6, 7, 8, 9 are used for for training, subject 11 for validation, and subject 5 for testing.

Metric. The prediction error is quantified via the MPJPE error metric [79, 118], which considers the

displacement of the predicted 3D coordinates w.r.t. the ground truth, in millimeters, at a specific future frame t:

$$L_{\text{MPJPE}} = \frac{1}{V}\sum_{v=1}^{V} ||\hat{\boldsymbol{x}}_{vt} - \boldsymbol{x}_{vt}||_2. \tag{2.5}$$

It is worth noting that, following literature [39, 117, 118], the loss function differs from the test metric, Eq. 2.5; namely, the loss function considers the average of MPJPEs over the entire predicted sequence:

$$L_{\text{MPJPE}} = \frac{1}{VT}\sum_{t=0}^{T}\sum_{v=1}^{V} ||\hat{\boldsymbol{x}}_{vt} - \boldsymbol{x}_{vt}||_2$$

where, in accordance with Eq. 2.5, $\hat{\boldsymbol{x}}_{vt}$ and $\boldsymbol{x}_{vt}$ are the 3-dimensional vectors of a target joint j_v $(0 \leq v \leq V)$ in a fixed frame f_t $(0 \leq t \leq T)$ for the ground truth and the predictions, respectively.

2.5.1 Modelling choices of SeS-GCN

We review and quantify the impact of the modeling choices of SeS-GCN:

Efficient GCN baselines. In Table 2.2, we first validate the three different modeling approaches to efficient GCNs, namely space-time separable STS-GCN [164], depth-wise separable graph convolutions DW-GCN [95], and Sparse-GCN [163]. STS-GCN yields the lowest MPJPE error of 117.0 mm at a 1-sec forecasting horizon (2.4% better than DW-GCN, 4.8% better than Sparse-GCN) with the fewest parameters, 57.6k (ca. x4 less). We build, therefore, on this approach.

Deeper GCNs. It is a long-standing belief that Deep Neural Networks (DNN) owe their performance to depth [68, 107, 190, 198]. However, deeper models require more parameters and have a longer processing time. Additionally, deeper GCNs may suffer from over-smoothing [132]. Seeking both better accuracy and efficiency, we consider three pathways for improvement: (1) add GCN layers; (2) add MLP layers between layers of GCNs; (3) adopt depth-wise graph convolutions, which also add MLP layers between GCN ones (cf. Sec. 2.3.2).

As shown in Table 2.2, there is a slight improvement in performance with 5 STS-GCN layers (MPJPE of 115.9 mm), but deeper models underperform. Adding MLP layers between the GCN ones (depth of 5+5) also decreases performance (MPJPE of 125.2). By contrast, adding depth by depth-wise separable graph convolutions (STS-DW-GCN of depth 5+5) reduces the error to 114.8 mm. This may be explained by the virtues of the increased depth in combination with the limiting cross-talk of joint-time channels, which existing literature confirms [34, 95, 164]. We note that space-time and depth-wise channel separability is complementary. Altogether, this performance is beyond the STS-GCN performance (114.8 Vs. 117.0 mm), at a slight increase of the parameter count (70k Vs. 57.6k).

Sparsifying GCNs and the proposed SeS-GCN. Finally, we target to improve efficiency by model compression. Trends have reduced the size of models by reducing the parameter precision [149], by pruning and sparsifying some of the parameters [126], or by constructing teacher-student frameworks, whereby a smaller student model is paired with a larger teacher to reach its same performance [70, 102]. Note that the last technique is the current go-to choice in deploying very large networks such as Transformers [14].

We start off by compressing the model with sparse adjacency matrices by the approach of Sparse-GCN [163]. They iteratively optimize the learned parameters and the masks to select some (the selection occurs by a network branch, also at inference, cf. 2.3.2). As illustrated in Table 2.2, the approach of [163]

Table 2.3: MPJPE error in mm for short-term (400 msec, 10 frames) and long-term (1000 msec, 25 frames) predictions of 3D joint positions on Human3.6M. The proposed model achieves competitive performance with the SoA [117], while adopting 1.72% of its parameters and running ∼4 times faster, cf. Table 2.5. Results are discussed in Sec. 2.5.2.

	Walking		Eating		Smoking		Discussion		Directions		Greeting		Phoning		Posing	
Time Horizon (msec)	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*
DCT-RNN-GCN [117]	**39.8**	**58.1**	**36.2**	**75.5**	**36.4**	**69.5**	**65.4**	119.8	56.5	106.5	**78.1**	138.8	**49.2**	105.0	**75.8**	178.2
MSR-GCN [39]	45.2	63.0	40.4	77.1	38.1	71.6	69.7	117.5	**53.8**	**100.5**	93.3	147.2	51.2	**104.3**	85.0	174.3
STS-GCN† [164]	51.0	70.2	43.3	82.6	42.3	76.1	71.9	118.9	63.2	109.6	86.4	**136.1**	53.8	108.3	84.7	178.4
SeS-GCN (proposed)	48.8	67.3	41.7	78.1	40.8	73.7	70.6	**116.7**	60.3	106.9	83.8	137.2	52.6	106.7	82.6	**173.5**

	Purchases		Sitting		Sitting Down		Taking Photo		Waiting		Walking Dog		Walking Together		Average	
Time Horizon (msec)	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*
DCT-RNN-GCN [117]	**73.9**	**134.2**	**56.0**	**115.9**	**72.0**	**143.6**	**51.5**	**115.9**	**54.9**	108.2	**86.3**	**146.9**	**41.9**	**64.9**	**58.3**	**112.1**
MSR-GCN [39]	79.6	139.1	57.8	120.0	76.8	155.4	56.3	121.8	59.2	**106.2**	93.3	148.2	43.8	65.9	62.9	114.1
STS-GCN† [164]	83.1	141.0	60.8	121.4	79.4	148.4	59.4	126.3	62.0	113.6	97.3	151.5	49.1	72.5	65.8	117.0
SeS-GCN (proposed)	82.2	139.1	59.9	117.5	78.1	146.0	57.7	121.2	58.5	107.5	94.0	147.7	48.3	70.8	64.0	113.9

does not make a viable direction (STS-DW-Sparse-GCN) since the error increases to 115.7 mm and the parameter count to $122.4k$.

Reminiscent of teacher-student models, in the proposed SeS-GCN, we first train a teacher STS-DW-GCN, then use its learned parameters to sparsify the affinity matrices of a student STS-DW-GCN, which is then trained from scratch. SeS-GCN achieves a competitive parameter count $58.6k$ and the lowest MPJPE error of 113.9 mm, being comparable with the current SoA [117] and using only 1.72% of its parameters ($58.6k$ Vs. $3.4M$).

2.5.2 Comparison with the state-of-the-art (SoA)

In Table 2.3, we evaluate the proposed SeS-GCN against the three most recent techniques over a short time horizon (10 frames, 400 msec) and a long time horizon (25 frames, 1000 msec). The first, DCT-RNN-GCN [117], the current SoA, uses DCT encoding, motion attention and RNNs and, differently from other models, demands more frames as input (50 vs. 10). The other two, MSR-GCN [39] and STS-GCN [164] adopt GCN-only frameworks, the former adopts a multi-scale approach, the latter acts a separation between spatial and temporal encoding.

Both on Short- and long-term predictions, at the 400 and 1000 msec horizons, the proposed SeS-GCN outperforms other techniques [117, 164] and it is within a 1.5% error w.r.t. the current SoA [117], while only using 1.72% parameters and being ∼4 times faster than [117].

2.6 Experiments on CHICO

We benchmark on CHICO the SoA and the proposed SeS-GCN model. The two HRC tasks of human pose forecasting and collision detection are discussed in Secs. 2.6.1 and 2.6.2, respectively.

2.6.1 Pose forecasting benchmark

Here we describe the evaluation protocol proposed for CHICO and report comparative evaluation of pose forecasting techniques.

Evaluation protocol. We create the train/validation/test split by assigning 2 subjects to the validation (subjects 0 and 4), 4 to the test set (subjects 2, 3, 18 and 19), and the remaining 14 to the training set.

†Results for STS-GCN differ from [164], due to revision by the authors, cf. `https://github.com/FraLuca/STSGCN`.

Table 2.4: MPJPE error in mm for short-term (400 msec, 10 frames) and long-term (1000 msec, 25 frames) prediction of 3D joint positions on CHICO dataset. The average error is 7.9% lower than the other models in the short-term and 2.4% lower in the long-term prediction. See Sec. 2.6.1 for a discussion.

	Hammer		High Lift		Prec. P&P		Rnd. P&P		Polishing		Heavy P&P		Light P&P		Average	
Time Horizon (msec)	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*
DCT-RNN-GCN [117]	41.1	**39.0**	69.4	128.8	50.6	83.3	52.7	88.2	42.1	76.0	64.1	121.5	62.1	104.2	54.6	91.6
MSR-GCN [39]	41.6	39.7	67.8	130.2	50.2	81.3	53.4	90.3	41.1	73.2	62.7	118.2	61.5	101.9	54.1	90.7
STS-GCN [164]	46.6	52.1	64.2	116.4	48.3	79.5	52.0	87.9	42.1	73.9	60.6	106.5	57.2	95.2	53.0	87.4
SeS-GCN (proposed)	**40.9**	49.3	**62.1**	**116.3**	**46.0**	**77.4**	**48.4**	**84.8**	**38.8**	**72.4**	**56.1**	**104.4**	**56.2**	**92.2**	**48.8**	**85.3**

For short-range prediction experiments, abiding the setup of Human3.6M [79], we consider 10 frames as observation time and 10 or 25 frames as forecasting horizon. Different from all reported techniques, DCT-RNN-GCN requires 50 input frames.

We adopt the same Mean Per Joint Position Error (MPJPE)[79] as Human3.6M, in Eq. (2.5), which also defines the training loss for the evaluated techniques.

None of the motion sequences for pose forecasting contain collisions. In fact, the objective is to train and test the "correct" collaborative human behavior, and not the human retractions and the pauses due to the collisions[**].

Comparative evaluation. In Table 2.4, we compare pose forecasting techniques from the SoA and the proposed SeS-GCN. On the short-term predictions, the best performance is that of SeS-GCN, reaching an MPJPE error of 48.8 mm, which is 7.9% better than the second best STS-GCN [164].

On the longer-term predictions, the best performance (MPJPE error of 85.3 mm) is also detained by SeS-GCN, which is 2.4% better than the second best STS-GCN [164]. The proposed model outperforms all techniques on all actions except *Hammer*, a briefly repeating action that may differ for single hits. We argue that DCT-RNN-GCN [117] may get an advantage from using 50 input frames (all other methods use 10 frames)

For a graphical illustration, Fig. 2.10 shows a distribution of the error per joint calculated over all the actions for the horizons 400 (*left*) and 1000 msec (*right*). In both cases, the error gets larger as we get closer to the extrema of the kinematic skeleton since those joints move the most. The slightly larger error at the right hand (70.03 and 125.76 mm, respectively) matches that subjects are right-handed (but some actions are operated with both hands).

For a sanity check of results, we have also evaluated the performance of a trivial zero velocity model. [120] has found that keeping the last observed positions may be a surprisingly strong (trivial) baseline. For CHICO, the zero velocity model scores an MPJPE of 110.6 at 25 frames, worse than the 85.3 mm score of SeS-GCN. This is in line with the large-scale dataset Human3.6M [79], where the performance of the trivial model is 153.3 mm.

2.6.2 Collision detection experiments

Evaluation protocol. We consider a collision to occur when any body limb of the subject gets too close to any part of the cobot, i.e. within a distance threshold, for at least one frame. In particular, a collision refers to the proximity between the cobot and the human in the forecast portion of the trajectory. The (Euclidean) distance threshold is set to 13 cm.

The motion of the cobot is scripted beforehand, thus known. The motion of the human subjects in the next

[**]After the collisions, the robot stops for 1 seconds, during which the human operator usually stands still, waiting for the robot to resume operations.

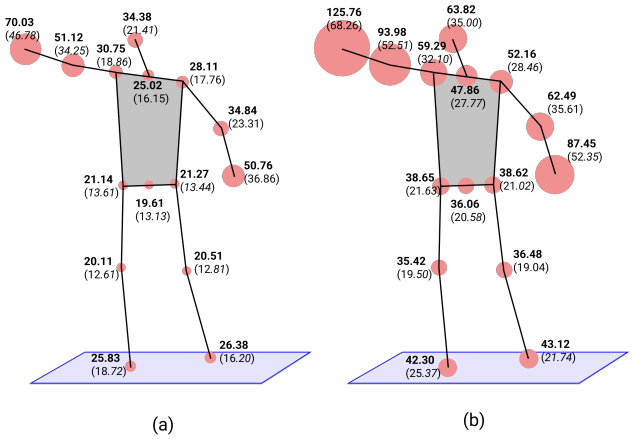


Figure 2.10: Average MPJPE distribution for all actions in CHICO on different joints for (a) short-term (0.40 s) and (b) long-term (1.00 s) predictions. The radius of the blob gives the spatial error with the same scale of the skeleton.

1000 msec needs to be forecast, starting from the observation of 400 msec. The train/validation/test sets sample sequences of 10+25 frames with stride of 10.

Evaluation of collision detection. For the evaluation of collision, following [124], both the cobot arm parts and the human body limbs are approximated by cylinders. The diameters for the cobot are fixed to 8cm. Those of the body limbs are taken from a human atlas.

In Table 2.5, we report precision, recall and F_1 scores for the detection of collisions on the motion of 2 test subjects, which contains 21 collisions. The top performer in pose forecasting, our proposed SeS-GCN, also yields the largest F_1 score of 0.64. The lower performing MSR-GCN [39] yields poor collision detection capabilities, with an F_1 score of 0.31.

Table 2.5: Evaluation of collision detection performance achieved by competing pose forecasting techniques, with indication of inference run time. See discussion in Sec. 2.6.2.

Time Horizon (msec)	***1000***			
Metrics	*Prec*	*Recall*	F_1	*Inference Time (sec)*
DCT-RNN-GCN [117]	0.63	0.58	0.56	9.1×10^{-3}
MSR-GCN [39]	0.63	0.30	0.31	25.2×10^{-3}
STS-GCN [164]	0.68	0.61	0.63	$\mathbf{2.3 \times 10^{-3}}$
SeS-GCN (proposed)	0.84	0.54	**0.64**	$\mathbf{2.3 \times 10^{-3}}$

2.7 Discussion

Towards the goal of forecasting human motion during human-robot collaboration in industrial (HRC) environments, we have proposed the novel SeS-GCN model, which integrates the three most recent modeling methodologies for accuracy and efficiency: space-time separable GCNs, depth-wise separable graph convolutions, and sparse GCNs. While SeS-GCN has demonstrated improved efficiency thanks to its novel design choices, further research is needed to understand if its application in real-world cases suffices to effectively increase safety and prevent collisions. In future work, we aim to assess the integration of such vision systems in simulated industrial processes by coupling a trajectory predictor model with a pose estimator. While an integration that guarantees real-time predictions has not been investigated yet, it would close the loop,

making the system fully autonomous. Also, we have contributed a new CHICO dataset, acquired at a real assembly line, the first providing a benchmark of the two fundamental HRC tasks of human pose forecasting and collision detection. Featuring an MPJPE error of 85.3 mm at 1 sec in the future with a negligible run time of 2.3 msec, SeS-GCN and CHICO unleash great potential for perception algorithms and their application in robotics.

Acknowledgements. This work was supported by the Italian MIUR through the project "Dipartimenti di Eccellenza 2018-2022", and partially funded by DsTech S.r.l.

Chapter 3

Contracting Skeletal Kinematics for Human-Related Video Anomaly Detection

3.1 Overview

Anomaly Detection (AD) is a broad and well-studied field in computer vision, which aims to detect events that deviate from normality automatically [29]. More precisely, the task is to detect anomalous events in footage and label the corresponding frames as abnormal. AD is a complex and multifaceted field with applications beyond just video surveillance [167]. Many techniques have been successfully applied in several real-world scenarios, including monitoring elderly individuals [141], industrial systems [6] and social networks [109].

While significant progress has been made in AD in recent years, this task still presents several challenges: (1) anomalous events are rare in real-world scenarios, and this reflects in the imbalanced distribution of normal and anomalous events in public AD datasets. (2) Anomalous events are challenging to identify since abnormal actions can involve either frenetic movements (e.g., fights) or very still postures (e.g., faints), resulting in a context-dependent definition of anomaly, which varies among public datasets. Furthermore, the same anomalous action can vary significantly among individuals, adding to the intra-class variation that must be considered. Finally, (3) when working with videos of humans, it is crucial to consider privacy and fairness concerns, such as avoiding violations of individuals' rights or exploiting social biases.

In this work, we propose COSKAD. This novel end-to-end model infers abnormality COntracting SKeletal (COSKAD) embeddings in the latent space and opens up a novel investigation of the latent metric space encoding the regular motions. The proposed method tackles all the limitations mentioned above. First, it adheres to the *One-Class-Classification* (OCC) [67, 108, 113, 119, 127] protocol, which simulates the scarcity of abnormal events exposed in (1). Moreover, COSKAD exploits the compact spatio-temporal skeletal representation (cf. Fig. 3.1) of human motion instead of raw frames, typically employed with video-based methods. This modality comes with several advantages; first, recent work [193] in the motion-related fields of Pose Forecasting and Action Recognition has proven the superiority of this modality with respect to the raw video frames for motion representation. Exploiting this input modality, our proposed model is more robust to slight motion deformations that represent the intra-class variability described in (2) while preserving the ability to distinguish between different actions, e.g., walking vs. running, which is crucial in AD. Furthermore, modeling agents as kinematic graphs allows for separating the person detection task (handled as a preprocessing step) from the anomaly detection task, resulting in a more computationally ef-

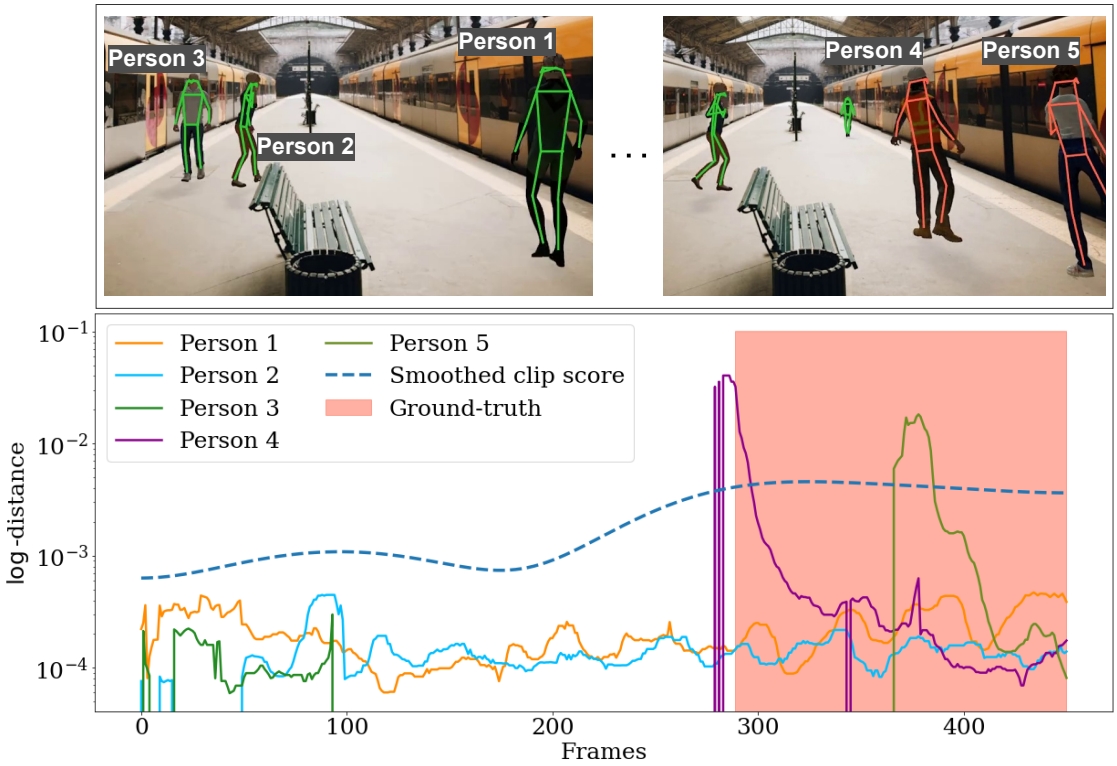


Figure 3.1: Anomaly score provided by COSKAD on a clip from the UBnormal dataset. COSKAD correctly classifies the motion of the two staggering characters (red skeletons in the upper-right picture) in the last part of the clip as anomalous.

ficient and context-agnostic method. Although skeletal representations are more compact than raw video frames, they contain the necessary information for effectively characterizing motions, as demonstrated by the state-of-the-art results of skeleton-based methods [113, 119, 127]. Finally, the skeletal representation is more privacy-preserving (3) since it ignores biometric details, representing all samples with anonymous tensors of coordinates. Consistently with this setting, we adopt the Human Related (HR) [127] split of the datasets, which corresponds to a version of the dataset that admits only anomalies generated by humans, e.g., people fighting, disregarding anomalies coming from the context scene, e.g., a car proceeding on the pavement.

The proposed model comprises two founding components: an encoder and a projector module. Differently from previous related works, which use either ST-GCN [112, 119, 194], or a Recurrent Neural Network [127] to encode the human motion, COSKAD implements its encoder as a Space-Time-Separable Graph Convolutional Network [164]. Similarly to the previous chapter, we find it beneficial to employ a separable GCN to limit the time-space information cross-talk, while the sparsity strategy achieved with the teacher-student framework is not effective in this context. This is probably due to the increased generality of motions encompassed by the Video Anomaly Detection datasets [2, 110, 112], making the sparsing strategy less effective. As far as we know, this is the first study to adapt it for skeletal-based AD and to expose a comparison among different GCN-based encoders (presented in Sec. 3.5.1). The projector module draws inspiration from SSL, where it has been shown to play a crucial role [33, 61], which we confirm here in detailed ablation studies (cf. Sec. 3.5.2). Finally, we define a data-driven metric objective in the latent space to minimize the distance between the skeletal embeddings and a center. We call the distance minimiza-

tion *contraction*, as it forces normality to concentrate around an origin. Dealing with a metric objective, we further propose a novel investigation of how the latent distribution might be altered, condensed, or expanded using the peculiar metric properties of three distinct manifolds: the Euclidean ($\mathbb{R}^n$), the hyperbolic space ($\mathbb{H}^n$), and the n-Sphere ($\mathbb{S}^n$). As far as we know, this is the first work to have studied different latent spaces for skeleton-based anomaly detection. The proposed model is simple, lightweight, and effective, as we illustrate in extensive experiments, where COSKAD outperforms state-of-the-art (SoA) models (including some video-based techniques) on three challenging benchmarks: *HR-ShanghaiTech Campus* [112, 119], *HR-Avenue* [110, 127], and the recent *UBnormal* [2].

Additionally, we offer an ablative examination of several of our model's key features. Beyond a thorough ablation study on the main modules of our proposed COSKAD, we also compare the encoder-based architecture proposed with an autoencoder and compare two alternative strategies to define the hypersphere center, extending the seminal study of [154]. Finally, we propose a novel HR version of UBnormal [2], dubbed *HR-UBnormal* as an additional contribution. We create HR-UBnormal by filtering out anomalous events that do not involve human individuals, e.g., we remove scenes of fire and car accidents unless people are involved. We leverage an established Pose Estimator [48] and refine its results with a Pose Tracker [192] to extract human poses in each UBnormal's video frame. So, HR-UBnormal contains human-related anomalies and human skeletons at all frames, inspected for accuracy and temporal consistency. To summarize, the contribution of this paper is threefold:

- We introduce COSKAD, a simple, end-to-end, and effective model that surpasses SoA results on three public benchmarks.
- We conduct an in-depth study on three different manifolds as latent spaces and explore their intrinsic properties, and thoroughly analyze their effects on our novel AD system.
- We introduce a new *HR* version of UBnormal with a filtered selection of clips featuring human-related events and an extended set of human body pose annotations.

3.2 Related Work

Anomaly Detection (AD) is a multi-faceted field with applications in several domains (see [19, 69] for surveys). This work focuses on skeleton-based anomaly detection, a type of video-based AD that involves analyzing the movements and poses of human bodies in a video. In this section, we compare works most closely related to ours, distinguishing error-based and score-based video AD techniques and skeleton-based models.

3.2.1 Video AD

Early proposed methods analyze the trajectories of agents in the video to unearth those that differ from normality [25, 84]. More recent deep learning methods for Video AD can be roughly collected into two categories: error-based or score-based.

Error-based methods. These methods attempt to detect anomalies through a generative process in which a model produces new video frames, which are then compared with ground truth. These methods assume that a model trained with only normal data will struggle to generate the anomalous frames, producing a more significant error that can be directly used as the anomaly score. [67] used convolutional AutoEncoder

(AE) and defined the input volume as a stack of sequential grayscale frames. [59] during the training phase builds a memory of the most representative normal poses. During the inference phase, for each sample, it finds the most similar example in memory and estimates its ground truth distance. Unlike these methods, COSKAD relies on a GCN-encoder, an ideal tool for exploring relationships between body joints over time in the kinematic graph and extracting semantically consistent latent embeddings.

Score-based methods. These approaches have been extensively studied [154, 171, 184]. They derive abnormality in videos by monitoring some quantity extracted from the embeddings produced by the deep network. For example, [156] proposed a two-stage method in which videos are divided into cubic patches and first analyzed with Gaussian classifiers to exclude the least relevant patches, e.g., background. Then, the remaining candidates are processed by a more complex CNN. In contrast, COSKAD looks at the latent space positions occupied by input embeddings to derive clues of abnormality. While working with images rather than video, Deep Support Vector Data Description (DSVDD) [154] and OC4Seq [184] are two methods related to COSKAD, as both employ a DSVDD objective seeking to minimize a sphere enclosing the generated embeddings. While we also employ an SVDD objective, our model learns in an end-to-end way to map the representation of the normal samples in the latent space while encoding semantic representations thanks to its separable GCN encoder. As far as we know, this work is the first to propose the SVDD objective for Video AD with skeleton-based representations.

3.2.2 Skeleton-based AD

[127] first introduces the skeleton-based representation in AD; their method presents a two-branches architecture for reconstruction and forecasting modeled as GRUs. [113] sets up the problem as motion forecasting and defines their model by stacking layers of ST-GCN [194] followed by an MLP forecasting module. While they introduce the use of GCN, the adjacency matrix is fixed and does not allow the exploration of intra-frame and intra-joint relationships, improving spatio-temporal features encoding [164]. [119] proposes a two-stage network in which they train an autoencoder (built on ST-GCN [194]), and, in the second stage, it clusters the produced embeddings in the latent space. These clusters should represent the normality styles in the train set, but it is challenging to spot the optimal number of clusters. Differently, COSKAD has an end-to-end approach and solves the problem of the number of clusters by forcing the entire train set into the same latent region, constraining the distances to a common center.

3.3 Methodology

In this section, we describe our proposed model focusing on its modules and the steps taken to train and assess it.

We assume the human body kinematics to be available as skeleton representations for a few given frames (cf. Sec. 3.4.1 for the details on the skeleton sequence extractions for the proposed HR-UBnormal dataset). These spatio-temporal graph inputs are fed to COSKAD which, as illustrated in Fig. 3.2, relies on two key components: a separable graph encoder and a projection module. The encoder processes the input graph and produces embeddings representing the motion of each individual. The projector adapts the embedding provided by the encoder for the mapping in the latent space. Both modules are jointly trained with a spatial minimization objective, which aims to catch the correspondences among samples belonging to the same class. Finally, we define a novel metrical objective in the latent space in order to guide the training and

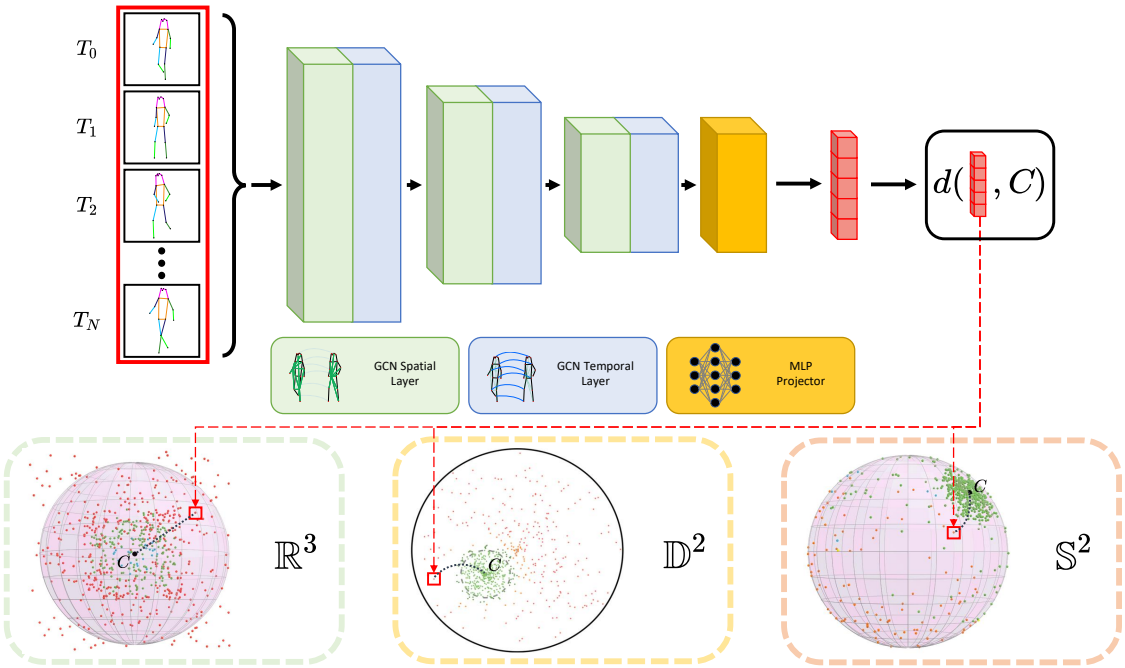


Figure 3.2: The overall architecture of COSKAD. The model combines an STS-GCN-based [164] encoder (light green and light blue blocks) with a projector module (yellow block) After projection, the latent representation (red vector in the figure) is embedded into the latent space. We propose and evaluate 3 variants of the latent space: *Euclidean* $\mathbb{R}^n$, *spherical* $\mathbb{S}^n$, and the *hyperbolic* modeled with the Poincaré Ball $\mathbb{D}^n$. During training, the embeddings are constrained to accumulate in a narrow region in the chosen manifold by reducing the distance between the motion embedding and the common center. The sequences mapped further from the center are interpreted as anomalous during inference.

consider three different manifolds as latent spaces: the *Euclidean Space* ($\mathbb{R}^n$), the *Poincaré Ball* ($\mathbb{D}^n$), and the *n-Sphere* ($\mathbb{S}^n$), to inherit their specific metric properties.

Formulation. The motion trajectories consist of V joints per actor in each frame tracked across all the frames (T_{actor}) where the actor is present. We apply a temporal sliding window crop on the trajectories to get sequences of V joints' spatial positions for T adjacent time frames. Finally, we organize the input signal as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with TV nodes $x_i \in \mathbb{R}^C$, where $C = 2$ stands for the x, y joint coordinates, and with edges $(i, j) \in \mathcal{E}$, represented by a spatio-temporal adjacency matrix $A^{st} \in \mathbb{R}^{VT \times VT}$, relating all joints to all others across all observed time frames.

3.3.1 Encoder and Projector module

We encode the body kinematics of a person with a SoA variant [164] of Graph Convolutional Network (GCN) [92]. GCNs are the go-to choice in the kinematic-related fields of Pose Forecasting [164, 193, 203] and action recognition [166, 194]. COSKAD leverages a separable GCN-encoder designed to capture spatio-temporal signals and produce consistent features. Specifically, the encoding performs a factorization of the adjacency matrix A into two learnable submatrices (A_s, A_t) responsible for spatial and temporal inter-connections, respectively. $A_s \in \mathbb{R}^{T \times V \times V}$, the *spatial adjacency matrix*, learns the relationships between different joints in each frame by learning their interdependence. In contrast, $A_t \in \mathbb{R}^{V \times T \times T}$, the *temporal adjacency matrix*, deals with the connections between different temporal instants for each joint.

This strategy, also adopted by [164] in the context of pose forecasting, ensures effective encoding of the spatio-temporal features of the input graph by learning and exploiting the joint-joint and time-time relation-

ships that characterize human motion. In addition, this factorization results in a considerable reduction in the number of parameters ($\sim$ 4x with respect to a plain GCN) since it does not consider the relationships between different joints in different frames.

We formulate a single encoder layer as:

$$X^{l+1} = \sigma(A_s A_t X^l W) \tag{3.1}$$

where X^l is the input from the previous layer, $W \in \mathbb{R}^{C \times C'}$ are learnable weights and σ an activation function. We stack four separable GCN layers interleaved with residual connections to encode the entire input sequence. Overall, the results of our ablation studies (cf. Sec. 3.5.1) suggest that the separable GCN is the best encoder among the GCN architectures we evaluated.

Reminiscent of recent works in Self-Supervised Learning (SSL) [33, 61], we explicitly define a projector module to refine the Encoder representation and accommodate it in the latent space. The projector comprises two identical blocks that iteratively process their input with a Fully Connected Module followed by a ReLU non-linearity and a Batch Normalization [78] layer. Despite its simplicity, we found it beneficial to add this module, as shown in Sec. 3.5.2.

3.3.2 Objective

The OCC formulation requires that the train sets contain elements of a single class. Therefore, the design of the objective function is crucial, as it can only exploit the similarities that normal samples exhibit. Many existing methods [59, 108, 113, 127] formulate the anomaly score using a proxy task, such as the reconstruction error. In Sec. 3.5.4, we experimentally confirm that this is suboptimal, as maintained in [55], since it does not align directly with the inference AD objective.

In this work, we define a metric objective that encourages the model to arrange the train samples in a narrow region in the latent space: inspired by [154], the objective is defined as:

$$\min_{\mathcal{W}} \frac{1}{N} \sum_{i=1}^{N} d(\Phi(x_i, \mathcal{W}),\ c) + \alpha f(\mathcal{W}) \tag{3.2}$$

Where Φ represents the mapping in the latent space defined by COSKAD, $\mathcal{W}$ is the set of its parameters, $x_i \in \mathbb{R}^{T \times V \times C}$ is an input sample, d is a metric defined in the latent space, f is a weight decay regularizing function, and c is the center of the hypersphere. We follow [154] and initialize c by taking the average of the final embeddings after an initial forward pass. Then, [154] suggests training the model with the parameter c fixed. We argue that this practice may harm the training, leading to a non-optimal solution since the center is precalculated and its position does not evolve with the learning.

Differently from [154], we introduce a novel and more tailored data-driven dynamic for the center: at the beginning of each training epoch, the position of the center is refined to be the centroid of the data's projection in the latent space. The benefit of this moving center is twofold. First, a data-driven center relieves the encoder and the projector learning not being constrained to accumulate projection around a fixed point. Secondly, it encourages COSKAD to explore the latent space to find a region that accommodates the representations, which we show to be beneficial in Sec. 3.5.3, especially when the hyperbolic manifold is set to be the latent space.

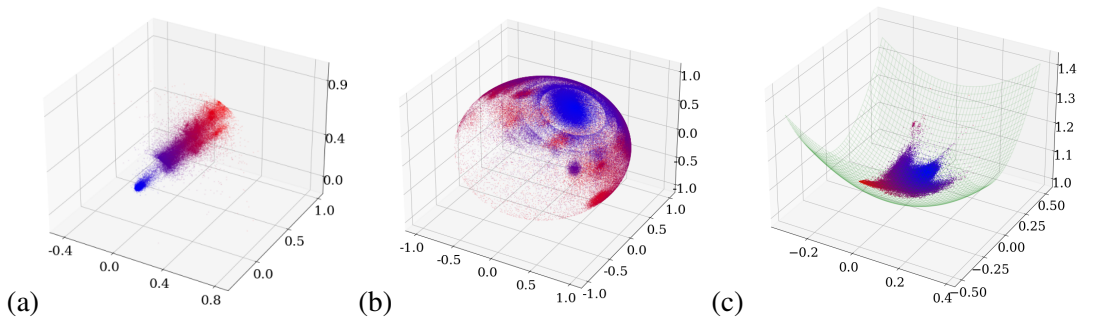


Figure 3.3: Visualization of the UBnormal test set's latent vectors embedded in three different manifolds: (a) Euclidean, (b) spherical, and (c) hyperbolic. We retain the three dimensions with the highest variance and color-code the points according to their distance from the center, from blue (closest) to red (furthest). Distance is intended as the L^2 norm in the Euclidean case, the *cosine distance* on $\mathbb{S}^n$, and the *Poincaré distance* for the hyperbolic embeddings. In the hyperbolic case, we highlight in green the hyperboloid onto which the embeddings are projected for better visualization.

3.3.3 Latent Spaces

We propose to consider three diverse manifolds for embedding the input sequences. In fact, the objective of Eq. 3.2 forces the model to focus on features corresponding to common characters in the embeddings extracted from the encoder and to iteratively reduce their distances from a center. As a result, the method relies on a metric which, in turn, depends on the metric space $\mathcal{L}$ chosen as the latent space. Motivated by this, we are the first to define the same objective on three manifolds, each with its own specific metric: the *Euclidean space* $\mathbb{R}^n$, the *spherical space* $\mathbb{S}^n \subset \mathbb{R}^{n+1}$, and the *hyperbolic space* $\mathbb{H}^n \subset \mathbb{R}^{n+1}$. These spaces have a different curvature that causes the distance to behave differently in each manifold.

Euclidean Latent Space

When $\mathcal{L} = \mathbb{R}^n$, the distance coincides with the L^2 metric $d_E(x, y) = ||x - y||_2$, and the model is defined substituting the distance d with d_E in Eq. 3.2.

n-Sphere Latent Space

With $\mathcal{L} = \mathbb{S}^n$, we modify the proposed COSKAD model, building on top of the *S-VAE* presented in [40]. We dub the model *COSKAD-radial* since it constrains inputs onto the spherical surface with a fixed radius $\mathbb{S}^n = \{x \in \mathbb{R}^{n+1} : \|x\|_2 = 1\}$; the posterior of the normal data approximates a Power Spherical distribution $q_X(x; \mu, \kappa)$ [41], with a constraint $L_{dir} = \frac{1}{N}\sum_{i=1}^{N}(1 - x \cdot c)$ to push the samples close to the empirical mean direction c. The target loss is $L = \gamma L_{rec} + \phi L_{dir} + \beta L_{KL}$, where L_{rec} defines the objective for the Variational AutoEncoder reconstruction, L_{KL} is the Kullback-Leibler divergence, $\gamma, \phi, \beta \in \mathbb{R}$ hyperparameters. The anomaly score is solely given by the *cosine distance* between any sample x on $\mathbb{S}^n$ and the empirical mean c computed at train time.

Hyperbolic Latent Space

When $\mathcal{L} = \mathbb{H}^n$, we model it with the *Poincaré Ball* which coincides with the unit euclidean open ball $\mathbb{D}^n$ endowed with the Riemmanian metric

$$g_x^{\mathbb{D}} = \frac{2I^n}{1 - ||x||^2},$$

where I^n represents the Identity matrix. This metric tensor induces the distance

$$d_{\mathbb{D}}(x,y) = \operatorname{arcosh}\left(1 + \frac{2||x-y||^2}{(1-||x||^2)(1-||y||^2)}\right). \tag{3.3}$$

The distances between the points on this manifold increase exponentially with distance from the origin O. To take full advantage of this property, we let the cluster center move in the Poincaré Ball in a data-driven fashion, but we also experienced defining the center as a fixed point, as detailed in Sec. 3.5.3. In Fig. 3.3, comparing the Euclidean (a) with the hyperbolic (c) case, the effect of this distance is shown: in the Euclidean case, farther points tend to be more spread since the hyperbolic distance $d_{\mathbb{D}}$ ensures a stronger attraction towards the center, i.e. $d_{\mathbb{D}}$ grows exponentially with the distance from the center. In this case, the model is obtained from COSKAD by adding a projection layer $\exp_0 : \mathbb{R}^{n+1} \to \mathbb{D}^n$, which performs the exponential mapping to get the hyperbolic representation in $\mathbb{D}^n$. Thus, the objective defined in Eq. 3.2 becomes:

$$\min_{\mathcal{W}} \frac{1}{N}\sum_{i=1}^{N} d_{\mathbb{D}}(\exp_0(\Phi(x_i,\mathcal{W})),\ c) + \alpha f(\mathcal{W}). \tag{3.4}$$

3.3.4 Anomaly Score

At inference time, the anomaly score s for each sample x (representing a single agent in a time window composed by T frames $\{f_1, f_2, ..., f_T\}$) is defined as the distance of its COSKAD embedding from the center c:

$$s(x) = d(\Phi(x,\mathcal{W}), c). \tag{3.5}$$

To score a single frame $\bar{f}$, for each agent p, we first collect all the windows involving p in a set w^p, and we restrict this set picking only the windows containing $\bar{f}$ to a set $w^p_{\bar{f}}$. Then, we calculate the score through Eq. 3.5 and thus set the score for p at frame $\bar{f}$ to be the mean of those scores:

$$s^p(\bar{f}) = \frac{1}{|w^p_{\bar{f}}|}\sum_{x \in w^p_{\bar{f}}} s(x). \tag{3.6}$$

Finally, we get the score for a single frame $\bar{f}$ by a max pooling operation over all the agents present in the scene at frame $\bar{f}$:

$$s(\bar{f}) = \max_{p \in P_{\bar{f}}}\{s^p(\bar{f})\}, \tag{3.7}$$

where $P_{\bar{f}}$ is the collection of all the people present in the clip at frame $\bar{f}$.

3.4 Experiments

In this section, we evaluate the performance of COSKAD on the human-related versions of three established benchmarks against state-of-the-art skeleton-based methods. Next, we discuss how our approach relates to techniques that additionally employ appearance. The section is organized as follows: first, we illustrate the datasets and the metric (Sec. 3.4.1). Then, we describe the results of the experiments and discuss the performance (Sec. 3.4.2). The last part of this section provides details about the implementation and data preprocessing (Sec. 3.4.3).

Dataset	# frames					
	Total	*Training*	*Validation*	*Test*	*Normal*	*Abnormal*
Avenue [110]	43,499	28,175	-	15,324	25,891	17,608
STC [112]	317,398	274,515	-	42,883	300,308	17,090
UBnormal [2]	236,902	116,087	28,175	92,640	147,887	89,015
HR-Avenue [127]	42,883	28,175	-	14,708	25,891	16,992
HR-STC [127]	313,212	274,515	-	38,697	297,090	16,122
HR-UBnormal (*Proposed*)	234,751	116,087	28,175	90,489	147,887	86,864

Table 3.1: Overview of the three datasets chosen, CUHK Avenue, ShanghaiTech Campus, and UBnormal, as well as their human-related versions.

3.4.1 Benchmarks

The following sections describe the datasets and introduce the metric we use to score anomalous frames in a video sequence.

Datasets

UBnormal. UBnormal [2] is the largest and most recent dataset for frame-level video anomaly detection, the only one to provide *train-validation-test* splits that adhere to the *Open Set* protocol, i.e., the sets of anomalous events for train, validation, and test are disjoint. It consists of 19 clips, which encompass both normal and abnormal events, for each of the 29 diverse background scenes. UBnormal has several distinctive features. First, as illustrated in Table 3.1, it contains more anomalous frames than those presented in CUHK Avenue or ShanghaiTech Campus. Additionally, it has labels for each anomaly at both the frame and pixel level, with annotations describing the anomalous actions (e.g., jumping, sleeping, stealing). UBnormal exhibits a larger list of anomalous event types than previous Video Anomaly Detection datasets: it includes 22 categories of anomalies, in contrast to ShanghaiTech Campus and CUHK Avenue which only encompass 11 and 5, respectively. Additionally, it introduces a variety of objects, such as cars and bicycles, to both the train and the test sets to avoid the anomaly being detected because the object was not seen during the training phase. In contrast to other benchmarks, it is synthetic and its virtual 3D scenes are created with Cinema4D with heterogeneous 2D backgrounds (e.g., streets, train stations, and office rooms, with different viewpoints and lighting conditions). Since COSKAD is trained in the OCC framework, we extract only the normal sample poses from the training set. Conversely, we maintain the original validation and test split in this setting.

HR-UBnormal (*Proposed*). We propose HR-UBnormal as an extension of the original UBnormal dataset with kinematic motion representations and a selected set of anomalies that relate only to human behaviors. AlphaPose [48] is first used to extract the poses, and PoseFlow [192] is used to track the skeletons throughout each video. We then filter out the non-human related anomalies. We remove the sub-sequences in which the only anomalous object was not a person (e.g., a car) or the anomaly cannot be detected using only body poses (e.g., fire in the scene). See Table 3.2 for the list of deleted non-HR anomalous actions.

As a result, we leave the validation set unaltered while eliminating the frames 2.32% of the test set. Table 3.1 lists the total number of normal and abnormal frames. **CUHK Avenue.** The CUHK Avenue

Clip	Length	Discarded	% discarded	% abnormality
abnormal_scene_4_scenario_1_fire	451	152	33.70	100.00
abnormal_scene_4_scenario_2_fire	451	35	7.76	100.00
abnormal_scene_7_scenario_1_fire	451	356	78.94	100.00
abnormal_scene_7_scenario_4	451	361	80.04	0.00
abnormal_scene_7_scenario_5	451	451	100.00	0.00
abnormal_scene_12_scenario_1_smoke	451	260	57.65	100.00
abnormal_scene_17_scenario_1_smoke	451	115	25.50	99.70
abnormal_scene_23_scenario_1	451	22	4.88	37.76
abnormal_scene_28_scenario_2	451	108	23.95	43.73
abnormal_scene_29_scenario_2_smoke	451	283	62.75	0.00
abnormal_scene_29_scenario_4	451	8	1.77	56.43

Table 3.2: List of the clips from which some non-human sub-sequences have been discarded. The column *Length* reports the number of frames of the original clip. The columns *Discard* and *% discarded* report the number of cut frames and the percentage w.r.t. the original clip's length, respectively. Finally, the column *% abnormality* shows the percentage of anomalous frames w.r.t. the current clip length; notice that some clips originally contained only anomalous frames, hence, after removing some of them, the percentage of abnormality remains 100.0%.

dataset [110] contains 16 training videos and 21 testing videos with a total of 47 abnormal events recorded with different camera positions and angles. The HR-version [127] is obtained by removing frames where (1) the anomalous event is non-human, (2) the person involved is occluded, or (3) the main subject cannot be detected and tracked.

ShanghaiTech Campus. The ShanghaiTech Campus (STC) dataset [112] contains footage from 13 cameras around the campus with different light conditions and camera angles. It contains more than 300,000 total frames, and there are 130 abnormal events, some of which are not present in other datasets (e.g., chasing, brawling). The HR-version [127] is obtained by removing 6 out of 107 test videos where the anomalous event is non-human.

Evaluation metric

We score each frame in a video as mentioned in Sec. 3.3.4. Then we compare it with the ground-truth labels to compute the *Area Under the Curve* (AUC) score, following the previous literature in Video AD [2, 56, 108, 110, 119, 127].

3.4.2 Comparison with SoA

We compare COSKAD against relevant skeleton-based state-of-the-art techniques on the HR version of the selected three datasets. This comparison, considering skeleton-based anomaly detection algorithms, is described in Sec. 3.4.2 and regards the bottom part of Table 3.3. Then, Sec. 3.4.2 compares skeleton-based Vs. video-based techniques, and it discusses progress on the two fronts.

	Video-based methods	Params	UBnormal	STC	Avenue
S	Sultani et al. [167]	-	50.3	-	-
	Georgescu et al.[56]	$>$ 80M	61.3	-	-
	Bertasius et al.[16]	121M	68.5	-	-
WS	Georgescu et al. [56]	$>$ 80M	59.3	82.7	92.3
OCC	Hasan et al. [67]	-	-	70.4	80.0
	Park et al. [134]	-	-	69.8	82.8
	Liu et al. [108]	14M	-	72.8	85.1
	Chang et al. [30]	-	-	73.3	86.0
	Barbalau et al. [9]	$>$ 80M	62.5	83.8	93.7
	Skeleton-based methods	**Params**	**HR-UBnormal**	**HR-STC**	**HR-Avenue**
OCC	Morais et al. [127]	25K	61.2	75.4	86.3
	Markovitz et al. [119]	805K	55.2	74.8	58.1
	Luo et al. [113]	8M	-	<u>76.5</u>	<u>87.3</u>
	Ours - *radial*	285K	63.4	75.2	82.2
	Ours - *Euclidean*	240K	<u>65.2</u>	**77.1**	**87.8**
	Ours - *hyperbolic*	240K	**65.5**	75.6	<u>87.3</u>

Table 3.3: Results on the human-related versions of the datasets UBnormal (*HR-UBnormal*), ShanghaiTech Campus (*HR-STC*) and CUHK Avenue (*HR-Avenue)*, measured in terms of AUC score (bottom part of the table). We highlight in bold the best results and underline the second best. We report the results of video-based models on the non-HR versions of the aforementioned datasets (upper part of the table); it should be noted that such methods cannot be directly compared with skeleton-only ones, which are rather complementary, and hence are displayed in gray. The blocks split the table according to each method's framework, where *S*, *WS* and *OCC* stand for *Supervised*, *Weakly-Supervised* and *One Class Classification* methods, respectively.

Comparison with skeleton-based techniques

Table 3.3 reports the results of the experiments we have conducted on HR-UBnormal, HR-ShanghaiTech Campus, and HR-Avenue. On HR-UBnormal, all three proposed volume shrinking strategies of COSKAD introduced in Sec. 3.3.3 outperform the current best [127]. In particular, our best variant, COSKAD-*hyperbolic*, outperforms [127] by a relative improvement of 7% (65.5 AUC Vs 61.2 AUC). This remarks on the effectiveness of COSKAD in the case of the challenging open set anomalies of UBnormal.

The HR-ShanghaiTech Campus and HR-Avenue rankings are tighter; still, COSKAD-*Euclidean* attains the best performance on both datasets, setting a new state-of-the-art on these benchmarks. Although the performance gain between COSKAD-*Euclidean* (77.1 and 87.8) and the current best skeleton-based method of [113] (76.5, 87.3) is relatively small, it should be noted that, even compared to skeleton-based baselines, our model remains lightweight (cf. the parameter count in Table 3.3). Comparing the three COSKAD versions, the *radial* approach underperforms. This might be a consequence of the more dispersed distribution of the embeddings from the center, as visible in Fig. 3.3.

Relation of COSKAD with video-based methods

For completeness, the upper part of Table 3.3 reports video-based techniques, tested on the complete set of videos of *UBnormal*, *ShanghaiTech Campus* and *CUHK Avenue*. The complete datasets include human-related anomalies, as well as anomalies not relating to people (e.g., crashing cars) and only stemming from

visual cues, such as fire, smoke, and fog. The general video-based anomaly detection techniques in the top part of Table 3.3 have access to the spatio-temporal volume of RGB-pixel information, accounting for appearance (color, shape, object-like, etc.) and motion cues (e.g., optical flow). On the other hand, skeletal motion accounts for a small subset of the information contained in videos, but it condenses cues about the actions, avoiding misleading features such as lightning conditions or viewing directions. Indeed, even without the ability to detect non-human events, our model achieves 65.0 AUC (cf. Table 3.4). In contrast, the current state-of-the-art OCC video-based model [9] only scores an AUC of 62.5. Besides, COSKAD only uses a fraction of parameters of [9], namely 240K Vs. 80M (-99.7%). On *ShanghaiTech Campus* and *CUHK Avenue*, our best-performing model reports an AUC score of 74.3 and 85.7, respectively, and only [9, 55] surpass COSKAD's performance. Overall, our models' high performance and reduced complexity suggest that skeleton-based techniques could be a valuable research direction to complement appearance-based cues. Moreover, by solely processing motion, as COSKAD does, additional privacy and fairness guarantees are provided, as the model cannot exploit biases related to skin color, gender, or clothing style.

3.4.3 Experimental setup

Implementation details. We train our proposed COSKAD with PyTorch Lightning using two Nvidia P6000 GPUs for 80 epochs, with a learning rate of 0.0001 and ADAM [91] optimizer. The training phase required 1.5 hours, which is a fraction of the training time of [119, 127]. We consider $V = 17$ key points to represent a pose and divide each agent's motion history by adopting a sliding window procedure (each window has a length of $T = 12$ frames with stride 1 so that windows overlap).
Pose normalization. Since the 2D positions of the joints refer to the frame, we normalize the poses to make them independent of the spatial location, following [127]. For all the datasets, we perform an additional normalization stage by applying robust scaling to reduce the contribution of outliers, as also done in [127].

3.5 Ablation Studies

In this section, we report additional results and experiments that have guided us in building COSKAD. This study focuses on establishing the two main components of the proposed model and compares two strategies concerning the center update and the scoring method. All the experiments presented in this section are performed on the established UBnormal dataset and are conducted using the best-performer Euclidean and the hyperbolic versions of COSKAD. Notably, each model's variant with an STS-GCN encoder presented in this section outperforms all the OCC techniques reported in Table 3.3.

3.5.1 Encoder

As illustrated in Sec. 3.3.1, COSKAD is equipped with a separable GCN-Encoder to process the input graphs. The first to propose a GCN as a kinematic encoder in the context of anomaly detection was [113], while previous works rely on LSTM [73], such as [127]. For this reason, we compare our encoder with other established GCN architectures, such as a plain GCN [92] and the ST-GCN [194], which have been employed in [113, 119]. As reported in Table 3.4, the selected separable GCN attains the best results among competitors, showing an increase in performance of 9.8% and 10.2% over GCN [92] and 3.6% and 7.8% over ST-GCN [194] on the Euclidean and hyperbolic models, respectively. This result confirms that separating the kinematic adjacency matrix in its spatial and temporal parts allows for improved input

Encoder			Projector			Center Update		Hyperbolic	UBnormal	
GCN[92]	ST-GCN[194]	Sep. GCN	Identity	Linear	Non-Linear	Static	Dynamic		Valid.	Test
✓					✓		✓		68.9	59.1
✓					✓		✓	✓	69.1	59.0
	✓				✓		✓		68.4	62.6
	✓				✓		✓	✓	72.3	60.3
		✓	✓			✓			72.7	64.5
		✓	✓			✓		✓	72.0	63.6
		✓	✓				✓		71.3	64.7
		✓	✓				✓	✓	72.9	64.5
		✓		✓		✓			69.8	64.2
		✓		✓			✓	✓	71.1	64.7
		✓			✓	✓			74.5	64.6
		✓			✓	✓		✓	74.0	63.1
		✓			✓		✓		75.8	64.9
		✓			✓		✓	✓	**76.4**	**65.0**

Table 3.4: Ablation on the components of the proposed method COSKAD. Red checkmarks indicate the technical choices we implement in the final model. The results are attained on the UBnormal dataset.

# Blocks	Euclidean		Hyperbolic	
	Validation	Test	Validation	Test
0	71.2	64.1	72.4	63.7
1	75.8	64.9	**76.4**	**65.0**
2	75.7	64.8	74.01	64.6

Table 3.5: Ablation on the depth of the Non-Linear projector proposed (cf. Sec.3.5.2).

representations. Moreover, separating the adjacency matrix results in a significant reduction in parameters, yielding an encoder with only 31K parameters against 75K and 170K of {ST-GCN, GCN}-based encoders, respectively.

3.5.2 Projector

The importance of the projector when dealing with representations and metric objectives has been studied thoroughly in the field of SSL [33, 61]. Following [33], we compare three different definitions for the projector: the *Identity* (no projector), *Linear* (two linear layers), and *Non-Linear* (one block of linear layer, non-linearity, and Batch Normalization [78] followed by another linear layer) projectors. In Table 3.4, we empirically confirm the results of [33], showing that a non-linear projector provides a boost in performance ($+5.6\%$, $+0.54\%$ on validation and test, respectively) when compared to an Identity projector, i.e., directly using the output of the encoder. The Identity and Linear strategies provide similar results, showing that a non-linearity is needed to improve the representations in the latent space. As a further investigation, we also verify the deepness of the non-linear projector module. Specifically, we assess our model against different versions of it with a naive non-linear decoder without any block, performing only Batch Normalization, a ReLU activation and a linear layer, and a deeper projector involving two non-linear blocks followed by a linear layer. As depicted in Table 3.5, a single non-linear block considerably increases performance, especially in the hyperbolic setting ($+2\%$). Since the COSKAD's modules operate in Euclidean spaces, this result is not surprising and exposes the need for a projector when dealing with non-Euclidean latent spaces. Further, using more layers results in a slight degradation in performance when using either the Euclidean or the hyperbolic latent space (-0.2%, -0.6%, respectively).

Model	Training	Inference	UBnormal
COSKAD	L_{hyp}	s_{hyp}	64.9
COSKAD-AE	$L_{hyp} + L_{rec}$	s_{rec}	63.0
		s_{hyp}	64.1
		$s_{hyp} + s_{rec}$	63.4

Table 3.6: Performance evaluation of our proposed models with COSKAD-AE. AUC score is reported for the UBnormal dataset.

3.5.3 Center Update Strategy

Since the metric objective of our method seeks to minimize the distance between the latent representations and a point defined in the latent space, it is critical to choose the optimal rule to update the center position during training to achieve optimal mapping. [154] proposed to fix a point in the latent space and then perform training around it. We experiment with this method, dubbed *Static*, and compare it with a novel rule to update the center position, dubbed *Dynamic*, in Table 3.4. While it is unclear which position for the center represents the optimal choice, we devise the *Dynamic* strategy to give more flexibility to the system. Indeed, in the *Dynamic* strategy, the center position is initially defined as the centroid of the training sequences embeddings at the beginning of the training phase. During training we then refine its position with the same technique every 2 epochs.

As can be seen, the *Dynamic* strategy provides the best result, especially in the hyperbolic space, with an increase of 3% *wrt* to its *Static* counterpart, while the gain in the Euclidean space ($+0.4\%$) is more marginal. We expected a similar behavior since the metric defined within the Poincaré Ball induces a distance that grows exponentially with the radius, therefore, when the learning center is left free to move, the model can take advantage of it. On the other hand, with the *Static* strategy, it can be challenging for the model to minimize the distances between the embeddings and the fixed center.

3.5.4 COSKAD AutoEncoder

We also consider a multi-task learning objective for COSKAD coupling the original objective illustrated in Eq. 3.2 with a reconstruction error; to do this, we included an additional module that acts as a decoder, allowing COSKAD to reconstruct the original poses. Altogether, this makes our proposed model a GCN AutoEncoder, dubbed COSKAD-AE, and the new module has been obtained by reversing the GCN-Encoder. Table 3.6 compares the performance of COSKAD-AE, trained with both reconstruction and hypersphere loss, against our best model. Three methods for calculating the anomaly score are possible: either the reconstruction or hypersphere loss can be employed alone (named s_{rec} and s_{hyp}, respectively in Table 3.6), or the two scores can be combined ($s_{rec}+s_{hyp}$). The best performances are achieved by the only hypersphere score. This is probably due to the Separable GCN-Encoder that, separately learning the trajectories of single joints and unified poses for each timeframe, exposes improved generalization that affects s_{rec} and provides better reconstructions.

3.6 Limitations

In Sec. 5.5 we prove the significance of our proposed COSKAD in the context of skeleton-based Video Anomaly Detection. However, despite the promising performance revealed, specific challenges persist,

Figure 3.4: Examples of extracted poses in *HR-UBnormal*. The poses are correctly detected even in challenging conditions, e.g., different scales or unusual poses. See section *Sample of misestimated human poses* for discussion.

Figure 3.5: Examples of misestimations of the pose extractor in *HR-UBnormal*. Fig. 3.5(a) shows a pose that is not present in the scene, Fig. 3.5(b) is an example of a pose that is not detected. Fig. 3.5(c) is an example of a noisy pose estimation due to the scale of the subject and its partial occlusion. See section *Sample of misestimated human poses* for discussion.

primarily related to misestimated skeletal poses occurring in the datasets and those due to the model shortcomings. These issues are commented in the following Sec. 3.6.1 and Sec. 3.6.2, respectively.

3.6.1 Samples of misestimated human poses

Despite being extracted from a synthetic dataset, *HR-UBnormal* poses have been obtained using a system that yields effective outcomes. In Fig. 3.4, we have included 3 frames from the proposed *HR-UBnormal* in which Alphapose correctly extracts poses of agents in complex positions (Fig. 3.4(a)) or at very different scales (Fig. 3.4(b), 3.4(c)). Nevertheless, Alphapose is not error-free, and in this section, we discuss some limitations of our dataset. Fig. 3.5 illustrates some misestimations performed by Alphapose: in Fig. 3.5(a), a pose is incorrectly detected from the background near two perfectly estimated people, Fig. 3.5(b) shows a lying agent whose pose is not detected by Alphapose, probably due to the supine pose assumed, and in Fig. 3.5(c) it can be seen that the system detects a noisy pose due to the subject's scale and partial occlusion. While we have tried to mitigate this problem by removing many noisy poses, some failure cases are challenging to recognize because they may last only one or a few frames.

3.6.2 Sample of COSKAD shortcomings

As mentioned in the section above, we have released several videos demonstrating our method's effectiveness and efficiency in detecting anomalies in situations of varying complexity. However, our system has some faults, and we will examine a few of them here. Fig. 3.6 depicts two examples of our method's failure, one showing a false positive (Fig. 3.6(a)) and the other a false negative (Fig. 3.6(b)). In the first, the person standing is dancing, and the system does not consider it abnormal, as evidenced by the declining slope in the anomaly score graph below. On the other hand, we show in Fig. 3.6(b) how COSKAD struggles to be

accurate when normal and abnormal actions are interspersed. One of the agents in this scene starts running, stops, and then resumes running. The graph demonstrates that the anomaly score is still high even during the clip's regular portion.

3.7 Discussion

We have proposed a novel Skeleton-based anomaly detection method based on the minimization of latent vectors to a center, exploiting the properties of three different manifolds: Euclidean, hyperbolic, and spherical. We defined the minimization metrics and scoring and investigated the alterations in space induced by the different manifolds' metric tensors. By leveraging an STS-GCN encoder and coupling it with a loss that matches the OCC objective, COSKAD outperforms SoA models on established human-related benchmarks. On the most recent and challenging HR-UBnormal, all three versions of the proposed COSKAD reach SoA performance, which shows the representational power of our approach. While these results are encouraging, our proposed model still exhibits some limitations in some specific cases; thus, in future work, we aim to investigate other Anomaly Detection methods as error- or prediction-based approaches in the context of skeletal data. Also, as already stated in Sec. 2.7, more research is needed to efficiently integrate the pose estimator model with COSKAD aiming to deploy a fully autonomous HR-VAD system.

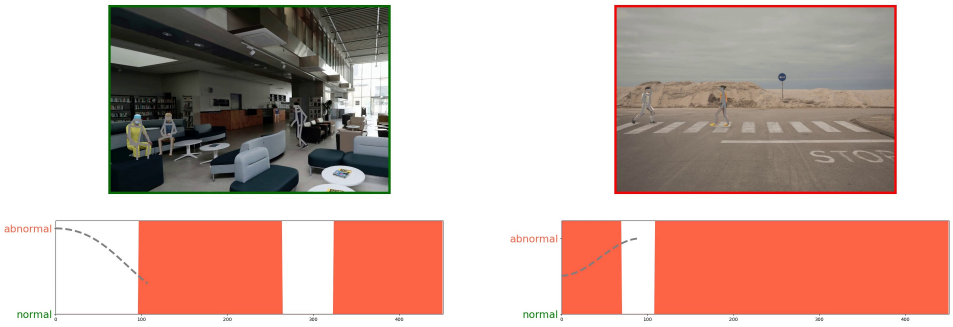


Figure 3.6: Examples of failure cases from the test set of *HR-UBnormal*, and the extracted score of the frame assigned by our proposed COSKAD. *(left)* the standing subject is dancing, but it is not detected as anomalous (*false negative*). *(right)* people depicted in the scene are walking, but the model predicts them as anomalous (*false positive*). See section *Sample anomaly detection and failure cases* for a broader discussion.

Chapter 4

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

4.1 Overview

Video Anomaly Detection (VAD) is a crucial task in computer vision and security applications. It enables early detection of unusual or abnormal events in videos, such as accidents, illnesses, or people's behavior which may threaten public safety [167]. However, several aspects make VAD a challenging task. Firstly, the definition of anomaly is highly subjective and varies depending on the context and application, making it difficult to define it universally. Secondly, anomalies are intrinsically rare. To account for data scarcity, models generally learn from regular samples only (also known as One Class Classification - OCC) or have to cope with the data imbalance. Thirdly, anomaly detection is intrinsically an open-set problem, and modeling anomalies must account for diversity beyond the training set.

An "ideal" model for anomaly detection should consider that there are infinitely many anomalous and non-anomalous ways of performing an action. Current state-of-the-art OCC techniques [51, 114, 119, 127] fail to address this issue. Indeed, they focus on learning either a single reconstruction or prediction of the input or deriving a latent representation of normal* actions, thereby constraining them to a limited latent volume. This last approach successfully accounts for the openset'ness of anomalies, i.e., anything mapped outside the normality region is considered abnormal. However, forcing normality into constrained volumes may not work for diverse-but-still-normal behaviors, i.e., OCC misclassifies as anomalous those not fitting in the volume.

We propose Motion Conditioned Diffusion Anomaly Detection - hereafter MoCoDAD - a novel generative model for VAD, which assumes that both normality and abnormality are multimodal†. Given a motion sequence, be it normal or anomalous, the sequence is split, and the later (future) frames are corrupted to become random noise. Conditioned on the first (past) clean input frames, MoCoDAD synthesizes multimodal reconstructions of the corrupted frames. The multimodal distribution of the future motions is achieved via multiple generations, effectively predicting diverse continuations of the motion. Indeed, given a single

*To avoid ambiguity, in this work, the term "normal" is the contrary of anomalous, not the synonym of "Gaussian". Normal refers to "normality" (or "normalcy"). Anomalous/abnormal refers to abnormality/anomaly.

†In this work, multimodal refers to distributions with multiple modes, not to mixing modalities (video, audio, text, etc.)

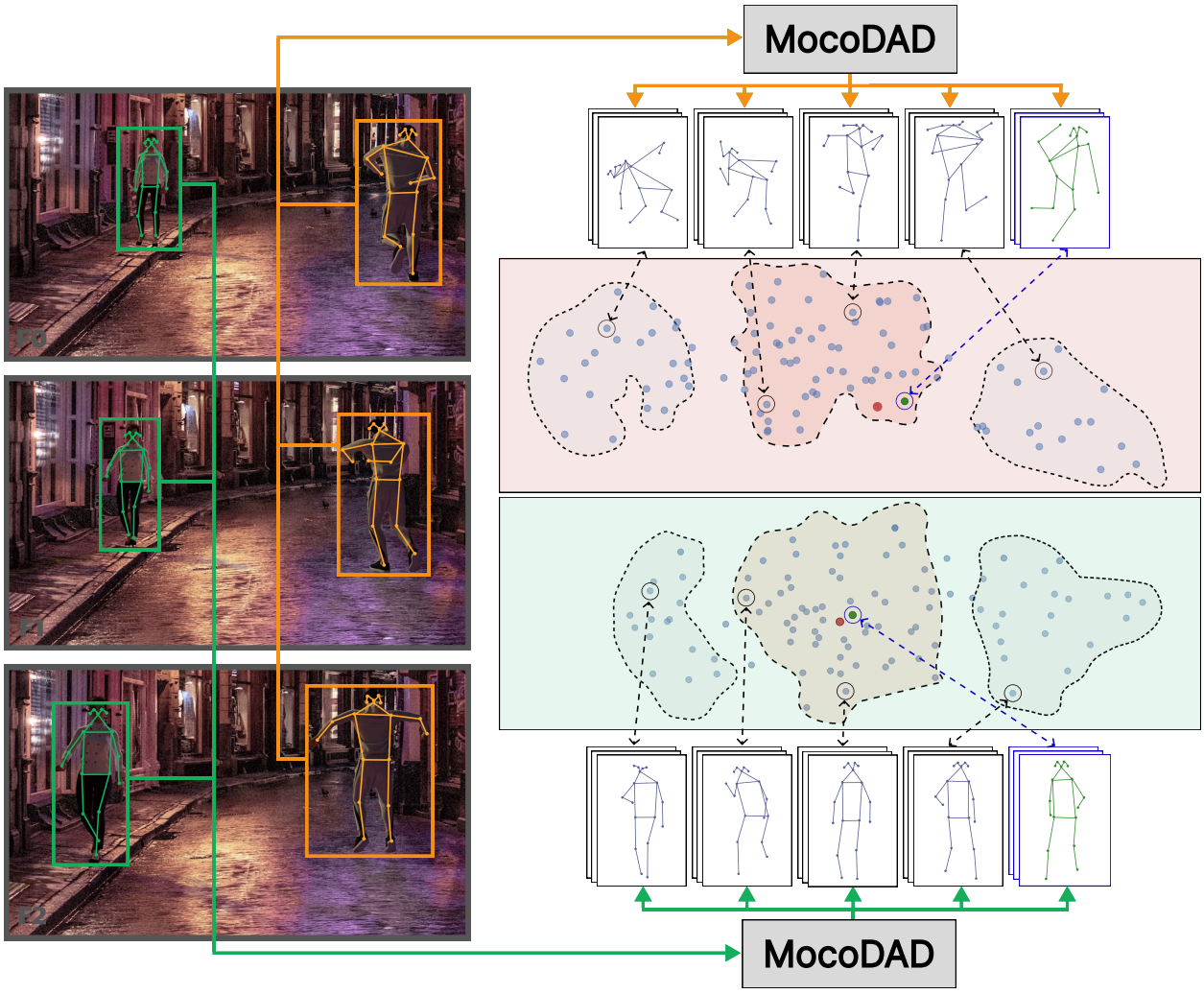


Figure 4.1: MoCoDAD detects anomalies by synthesizing and statistically aggregating multi-modal future motions, conditioned on past poses (frames on the left). Red (top) and green (bottom) distributions represent examples of anomaly and normality generations (2d mapped via t-SNE). Within the distribution modes (dashed-contoured), the red dots are the actual true futures corresponding to the conditioning past frames. 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.

"past" motion, we generate several diverse futures to account for intra-class variability. MoCoDAD then discerns normality from anomaly by comparing the multimodal distributions. In the case of normality, the generated motions are diverse but pertinent, i.e. they are biased towards the true uncorrupted frames. In the case of abnormality, the synthesized motion is also diverse, but it lacks pertinence, as shown in Fig. 4.1 and discussed in Sec. 4.6.

MoCoDAD is the first diffusion-based technique for video anomaly detection. We are inspired by Denoising Diffusion Probabilistic Models (DDPMs) [72, 165], state-of-the-art, among others, in image synthesis [146, 147, 153], motion synthesis [172, 191], and 3D generation tasks [206]. DDPMs are selected for their improved mode coverage [189], i.e. they generate diverse multimodal motions, which MoCoDAD statistically aggregates (see the respective ablative study in Sec. 4.6).

Besides multimodality, a crucial aspect of MoCoDAD is the choice of the conditioning strategy to guide the synthesis. We consider human motion as skeletal representations and propose corrupting the body joint

coordinates at each frame by displacing them with random translations. Conditioning refers to the process that provides the model with the uncorrupted first part of the motion sequence (past) to guide the denoising of the corrupted second part (future). In this study, we compare three modeling choices to find the most suitable conditioning strategy. We experiment with (1) directly feeding the denoising module with the concatenation of the uncorrupted poses with the displaced sequence [155]. For the other two strategies, we get a latent representation of the input (via (2) an encoding module and (3) an autoencoder) and feed the network with this learned representation (cf. Sec. 4.3.3). The third strategy works best (cf. Sec. 4.6.3).

We evaluate MoCoDAD on three challenging benchmarks of human-related anomalies, namely, HR-UBnormal [2, 51], HR-ShanghaiTech Campus [112, 127] (HR-STC), and HR-Avenue [110, 127], and on the most recent VAD dataset UBnormal [2]. MoCoDAD achieves state-of-the-art (SoA) performance on all four datasets, which demonstrates the effectiveness of modeling multimodality for normal and abnormal motions. Notably, by not using appearance, MoCoDAD benefits increased privacy protection (no visual facial nor body features) and better computational efficiency, thanks to the lightweight body kinematic representations. We summarize our contributions as follows:

- A novel generative VAD model based on comparing the multimodality of normal and abnormal motion generations;
- The first probabilistic diffusion-based approach for VAD, which fully exploits the enhanced mode-coverage capabilities of diffusive probabilistic models;
- A novel motion-based conditioning on the clean input sequence to steer the synthesis towards diverse pertinent motion in the case of normality;
- A thorough validation on UBnormal, HR-UBnormal, HR-STC, and HR-Avenue benchmarks where we outperform the SoA by 5.1%, 4.4%, $.5\%$, $.8\%$, respectively.

4.2 Related Work

Previous work relates to ours from two main perspectives: Video Anomaly Detection methods (see Sec. 4.2.1) and diffusion models for motion synthesis (see Sec. 4.2.2).

4.2.1 Video Anomaly Detection Techniques

Pioneer works analyze the trajectory of the agents in the frames to discriminate those distant from normality [25, 84, 99]. Within recent literature, two major trends can be identified: latent- and reconstruction-based methods. VAD techniques also vary based on the type of input data they use, such as videos or human skeletal pose motions. MoCoDAD, as all the VAD works presented in this section, adheres to the OCC protocol, which simulates the scarcity of anomalies in real-world scenarios [89].

Latent-based VAD methods identify abnormality according to a score extracted from a learned latent space whereby normality is supposedly mapped into a constrained volume, and anomalies are those latents lying outside, with a larger score (see [154, 158, 171, 184] for an overview of latent-based AD). Sabokrou et al. [156] propose a two-staged cascade of deep neural networks. First, they employ a stack of autoencoders that detects points of interest (POIs) while excluding irrelevant patches (e.g., background). Second, they identify anomalies by densely extracting and modeling discriminative patches at POIs. Notice that this work constrains normality to belong to a single mode and anomalies outside, thus, addressing the openset'ness of anomalies, but it hampers the multimodal and diversity [197] aspect of normal motions. Contrarily, our work considers the multimodality of normal and abnormal motions.

Notably, Nguyen et al. [129] propose an image-based technique exploring multimodal anomaly detection via multi-headed VAEs. However, considering a fixed number of modes for reality amends multimodality only partially, as it misses to unleash its openset'ness. Differently, we adopt diffusion models for their improved mode coverage and generate multiple futures, not being constrained on a fixed number of heads (see Sec. 4.6).

Reconstruction-based VAD methods consider the original metric space of the input and leverage reconstruction as the proxy task to derive an anomaly score. These models are trained to encode and reconstruct the input from normal events, producing larger errors on anomalies not seen during training. [35, 67, 201] use sequences of frames and feed them to convolutional autoencoders. Gong et al. [59] "memorize" the most representative normal poses to discriminate new input samples. Liu et al. [108] tackle intensity and gradient loss, optical flow, and adversarial training. Luo et al. [112] use stacked RNNs with temporally-coherent sparse coding enforcing similar neighboring frames to be encoded with similar reconstruction coefficients. Barbalau et al. [10] builds upon [56] and integrates the reconstruction of the input frames, via multi-headed attention, into a multi-task learning framework. Besides [56, 108], all works rely on a single reconstruction proxy task via non-variational architectures that learn discrete manifolds. However, normality and abnormality are multimodal and diverse, making it hard for these techniques to have an exact match (reconstruction) over the GT. Additionally, GANs used in [108] suffer from mode collapse [173] lacking to represent the multimodality of reality. Similarly to [129], [10] can represent only a fixed number of modalities, which does represent the openset'ness of reality. MoCoDAD is a reconstruction-based approach and leverages diffusion processes [43] to account for the openset'ness of normalcy and anomalies in terms of pertinence to the GT.

Skeleton-based VAD methods have already been commented in Sec.3.2. As a further work, we also mention COSKAD, which has been introduced in the previous chapter. Adhering to the DSVDD technique, COSKAD force the normal instances into the same latent region, driving the distances to a common center, and deeming as abnormal those samples that do not belong to the learned normality region. MoCoDAD is also a skeleton-based approach, but unlike COSKAD, it relies on an error-based technique also considering the inthrinsic multimodality of reality.

4.2.2 Diffusion Models

Diffusion models have marked a revolution in generative tasks such as image and video synthesis [155, 172, 191], but they have not been employed for VAD. Saadatnejad et al. [155] propose a two-step framework based on temporal cascaded diffusion (TCD). First, they denoise imperfect observation sequences and, then, improve the predictions of the (frozen) model on repaired frames. Tevet et al. [172] use a transformer encoder to learn arbitrary length motions [3, 139] coherent with a particular conditioning signal c. They experiment with constrained synthesis where c is a text prompt (i.e., text-to-motion) or a specific action class (i.e., action-to-motion) and unconstrained synthesis where c is not specified. Chen et al. [191] design a transformer-based VAE [139] to learn a representative latent space for human motion sequences. They apply a diffusion model in this latent space to generate vivid motion sequences while obeying specific conditions similar to [172]. Differently, MoCoDAD is a diffusion-based model that uses conditioning over a portion of the input (e.g., previous frames condition the generation of future ones).

Wyatt et al. [188] propose AnoDDPM, a diffusion model on images, which does not require the entire Markov chain (noise/denoise) to take place. They use decaying octaves of simplex noising functions to distinguish the corruption rate of low-frequency components from high-frequency ones. However, they

add and remove noise without conditioning, identifying anomalies when noise removal diverges from the input. Differently, MoCoDAD is based on generating and comparing multimodal motions against the GT in terms of pertinence. Our proposed model is the first to exploit the multimodal generative and improved mode-coverage capabilities of diffusive techniques, via forecasting tasks, further to being first in adopting them for detecting video anomalies. Hence, to transfer DDPMs from video-based Anomaly Detection to skeleton-based VAD we rely on a U-Net-shaped stack of STS-GCN [157, 164] layers, which includes the spatio-temporal aspects of joints in sequences of human poses.

4.3 Methodology

MoCoDAD learns to reconstruct the later (future) corrupted poses by conditioning on the first (past) poses. Sec. 4.3.2 describes the training diffusion denoising process, how it generates multimodal reconstructions, and statistically aggregates them at inference to detect anomalies. In Sec. 4.3.1 we briefly present preliminary concepts of DDPMs. Then, Sec. 4.3.3 details conditioning on past frames, and Sec. 4.3.4 describes the architecture of MoCoDAD. Finally, in Sec. 4.3.5 and 4.3.6 we provide the reader with a thorough description of the MoCoDAD algorithms and implementation details, respectively.

4.3.1 Background on Diffusion Models

A denoising diffusion probabilistic model (DDPM) [72, 195] 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 $t = 1, \ldots, T$, transforming any data distribution $q(x_0)$ into a simple prior (e.g., Gaussian). The forward process can be expressed as

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbb{I}) \tag{4.1}$$

To shift the data distribution $q(x_0)$ toward $q(x_t|x_{t-1})$ in one single step, equation 4.1 can be reformulated as

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbb{I}) \tag{4.2}$$

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

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\big(x_{t-1}; \mu_\theta(x_t, t), \beta_t\mathbb{I}\big) \tag{4.3}$$

where $\mu_\theta(x_t, t)$ is a deep neural network that estimates the forward process posterior mean. [72] has shown that one obtains high-quality samples when optimizing the objective

$$\mathcal{L}_{simple} = \mathop{\mathbb{E}}_{t,x_0,\varepsilon}\Big[\big|\big|\varepsilon - \varepsilon_\theta\big(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon, t\big)\big|\big|_2^2\Big] \tag{4.4}$$

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 [72] is used to iteratively denoise random Gaussian noise $x_T \sim \mathcal{N}(0, \mathbb{I})$, to generate a sample from the learned distribution. [147, 153] have shown that one may condition DDPMs on a signal h by feeding it to the neural network ϵ_θ.

4.3.2 Diffusion on Trajectories

Training We define a diffusion technique that learns to reconstruct corrupted future motion sequences conditioned on clean past ones.

Let $X_a = \{x_a^1, \dots, x_a^N\}$ be a sequence of N time-contiguous poses belonging to a single actor a. Since our model considers one actor at a time, we use $X = \{x^1, \dots, x^N\}$ for notation simplicity. Each pose x^i can be seen as a graph $x^i = (J, A)$ where J represents the set of joints, and A is the adjacency matrix representing the joint connections. Notice that each joint is attributed with a set of spatial coordinates in $\mathbb{R}^C$, hence $x^i \in \mathbb{R}^{|J| \times C}$, and $X \in \mathbb{R}^{N \times |J| \times C}$. Here, we use $C = 2$ as the person's pose is extracted from images at each frame.

We divide X into two parts: the past $X^{1:k}$ and the future sequence of poses $X^{k+1:N}$ with $k \in \{1, ..., N\}$.

During the forward process q, we corrupt the coordinates of the joints by adding random translation noise. We sample a random displacement map[‡] $\varepsilon^{k+1:N} \in \mathbb{R}^{(N-k) \times |J| \times C}$ from a distribution $\mathcal{N}(0, \mathbf{I})$ and add it to $X^{k+1:N}$ to randomly translate the position of its nodes.

The magnitude of the added displacement depends on a variance scheduler $\beta_t \in (0, 1)$ and a diffusion timestep $t \sim \mathcal{U}_{[1,T]}$. As a result, q increasingly corrupts the joints x^i at each diffusion timestep t (i.e., $x^i_{t=1} \rightarrow \cdots \rightarrow x^i_{t=T}$) making $x^i_{t=T}$ indistinguishable from a pose with randomly sampled joints' spatial coordinates.

The reverse process p_θ unrolls the corruption, estimating the spatial displacement map $\varepsilon^{k+1:N}$ via a U-Net-like architecture ε_θ (see Sec. 4.3.4 for more details). To achieve an approximation of $\varepsilon^{k+1:N}$, we train the network conditioned on the diffusion timestep t (embedded through an MLP τ_θ) and the embedding h of the previous trajectory $X^{1:k}$.

In Eq. 4.5, similarly to [153], we define the displacement estimation objective[§].

$$\mathcal{L}_{disp} = \mathbb{E}_{t,X,\varepsilon}\left[\left|\left|\varepsilon - \varepsilon_\theta(X_t, t, h)\right|\right|\right] \tag{4.5}$$

Inspired by [58], we smooth $\mathcal{L}_{disp}$ as follows:

$$\mathcal{L}_{smooth} = \begin{cases} 0.5 \cdot (\mathcal{L}_{disp})^2 & \text{if } |\mathcal{L}_{disp}| < 1 \\ |\mathcal{L}_{disp}| - 0.5 & \text{otherwise} \end{cases} \tag{4.6}$$

Inference At inference time, MoCoDAD generates multimodal future sequences of poses from random displacement maps, conditioned on the past frames, then aggregates them statistically to detect anomalies.

We sample a random displacement $z \sim \mathcal{N}(0, \mathbf{I})$ and consider it the starting point of the synthesis process that generates a future human motion via Eq. 4.7.

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(X_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}}\varepsilon_\theta(X_t, t, h)\right) + z\sqrt{\beta_t} \tag{4.7}$$

Note the motion conditioning h, encoding the past k frames. We generate m diverse future pose trajectories

[‡]In this paper, a displacement map is equivalent to the addition of noise (corruption) to the input, e.g., in [72]. We use this term to emphasize that *we move the joints away from their original spatial position*. We invite the reader to consider displacing and corrupting as interchangeable here.

[§]For readability purposes, for what follows, we omit the superscript $k + 1{:}N$, and assume that we are considering only future motion.

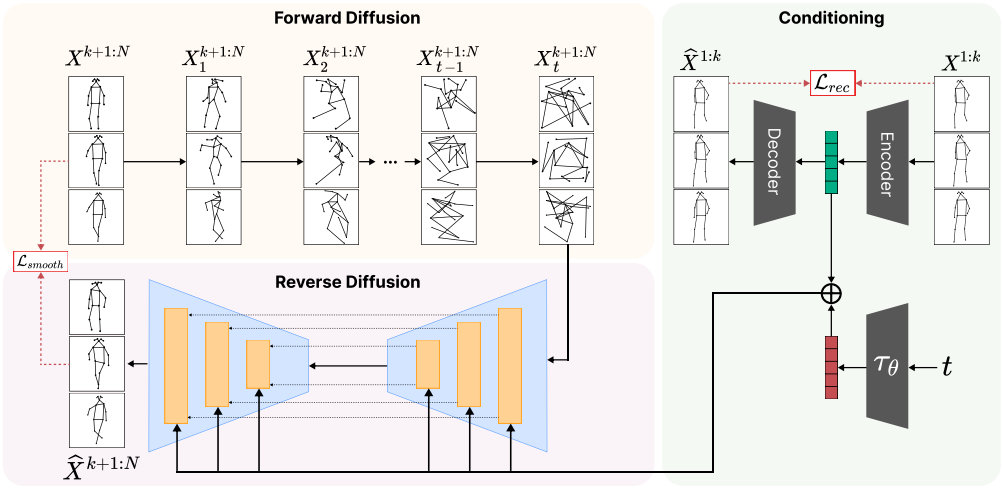


Figure 4.2: Overview of the proposed MoCoDAD. A sequence of N skeletal motions ($N = 6$ in the example) is split into past (top-right $X^{1:k}$ frames, $k = 3$ in the example) and future (top-left $X^{k+1:N}$ frames). During training, the Forward Diffusion block adds noise to the future frames, shifting each joint by a random vector displacement of varying intensity (increasing with the diffusion timestep t). Then the Reverse Diffusion learns to estimate the noise. A key aspect of MoCoDAD is the conditioning, i.e. how to encode the past clean k frames and guide the synthesis of relevant futures.

$Z_1, \ldots, Z_m$. For each Z_i, we compute the reconstruction error via the smoothed loss $s_i = \mathcal{L}_{smooth}(|X - Z_i|)$ used in training (cf. Eq. 4.6).

We aggregate all the scores from the generations $S = \{s_1, \ldots, s_m\}$ to distill a single anomaly score for that sequence. This score is subsequently assigned to the corresponding frames to assess their anomaly level. In scenarios where multiple actors occupy the same frames, we assign the average anomaly score to those frames. We explore different strategies for distilling the anomaly score from S: (1) the diversity between the normal and anomalous generations and (2) aggregation statistics. To account for diversity, we consider the diversity metric $rF = \text{mean}(\mathcal{L}_{smooth}) / \min(\mathcal{L}_{smooth})$ introduced by [26, 135]. Diversity stems from testing whether anomalous generations are more diverse than normals, but the diversity score fails to detect anomalies, nearly dropping to random chance. We explain this by normalcy and anomaly having a similar degree of diversity, as we experimentally validate in Sec. 4.6.1.

We consider as aggregation statistics the mean, quantile robust statistics including the median, as well as maximum and minimum selectors, in terms of $\mathcal{L}_{smooth}$ distances between the m generated and the GT future motion. Our analysis highlights that the minimum distance is the best on average. This reinforces the original hypothesis that normality-conditioned generated motions are as diverse as abnormal-conditioned ones but more biased to the actual motion, thus more likely to generate samples close to it. See Sec. 4.6.2 for the experimental evaluation.

4.3.3 Motion Conditioning for multimodal Pose Forecasting

The choice of the conditioning strategy is a crucial factor for diffusion models, as it determines how the conditioning information is fed into the network, and it directly affects the quality of the outputs. In this work, we propose a thorough examination of different strategies for feeding the diffusion models with the conditioning information.

We identify three different modeling choices for conditioning the diffusion, illustrated in Fig. 4.3, i.e.,

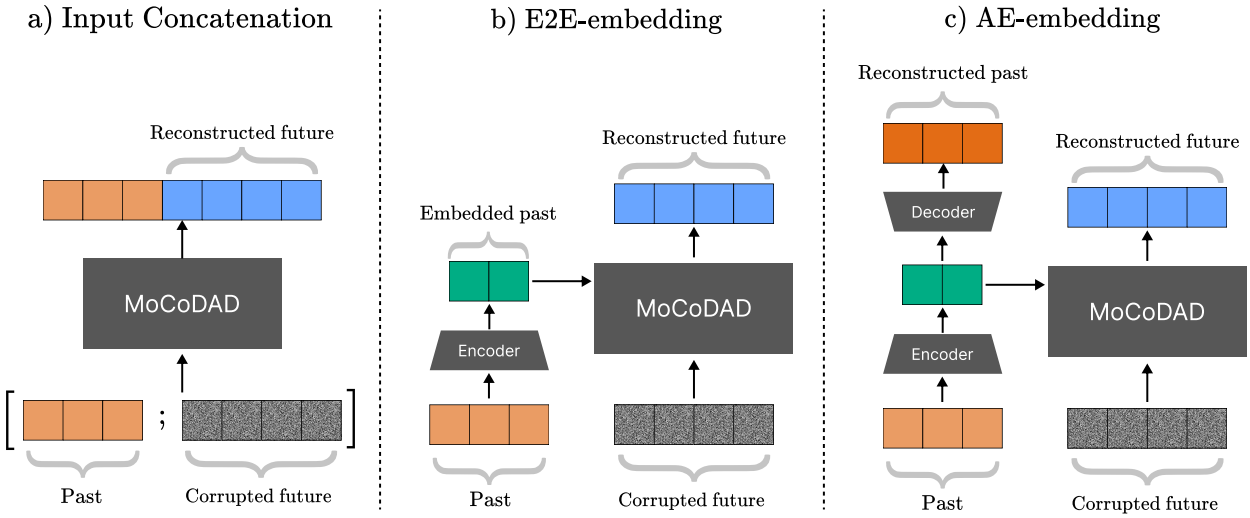


Figure 4.3: Comparison of the three conditioning strategies.

input concatenation, *E2E-embedding*, and *AE-embedding*. *Input concatenation* refers to conditioning with a portion of the raw input motion. Here, we keep the past sequence of poses $X^{1:k}$ (the conditioning signal) uncorrupted and prepend it to the corrupted future sequence $X_t^{k+1:N}$. Input concatenation has been explored in previous work for pose forecasting reaching SoA performances in [155].

Both *embedding* choices refer to passing the conditioning past frames through an encoder E, then providing them to all latent layers of the denoising model (cf. incoming vertical arrows into the orange layers in Fig. 4.2). Here, E is a GCN [164] and it encodes sequences of poses $X^{1:k}$ into the representation $h = E(X^{1:k})$. In the case of *E2E-embedding*, E is jointly learned with the rest of the architecture, leveraging the training $\mathcal{L}_{smooth}$ loss. The *AE-embedding* adds an auxiliary reconstruction loss $\mathcal{L}_{rec}$ to support training E, as it tasks a decoding network D to reconstruct the conditioning past frames according to the following loss function:

$$\mathcal{L}_{rec} = \left|\left| D(E(X^{1:k})) - X^{1:k} \right|\right|_2^2 \tag{4.8}$$

In the case of *AE-embedding*, the auxiliary is summed to the main loss, resulting in the following total loss:

$$\mathcal{L}_{tot} = \lambda_1 \mathcal{L}_{smooth} + \lambda_2 \mathcal{L}_{rec} \tag{4.9}$$

where λ_1,$\lambda_2 \in [0, 1]$ account for the contribution of each loss function respectively. Lastly, since DDPMs benefit from being conditioned on the timestep t, we add the embedding $\tau_\theta(t)$ to the latent h and feed the resulting motion-temporal signal to each layer of our network ε_θ [172].

The results of the evaluation are discussed in Sec. 4.6.3, whereby applying AE-embedding conditioning on $X^{1:k}$ emerges with results beyond the SoA.

4.3.4 Architecture Description

Fig. 4.2 illustrates the architecture of MoCoDAD. We distinguish two main blocks: a conditioning auto-encoder for the past motion, $X^{1:k}$, and a denoising model for $X^{k+1:N}$. The main diffusion model architecture is the neural network, represented with orange blocks, tasked with estimating the corrupting noise in the input motion, thus reconstructing the actual future motion. As done in [188], we rely on a U-Net

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

$\overline{\mathcal{P}} = \mathrm{E}(\mathcal{P})\ ;\ \hat{\mathcal{P}} = \mathrm{D}(\overline{\mathcal{P}})$

$\tau = \tau_\theta(t)$

// Forward Diffusion

$\mathcal{F}_t = q(\mathcal{F}, t)$

// Engender futures

$\hat{\mathcal{F}} = \text{MoCoDAD}(\mathcal{F}_t; \tau, \overline{\mathcal{P}})$

// Loss

$Loss = \lambda_1 \mathcal{L}_{smooth}(\hat{\mathcal{F}}, \mathcal{F}) + \lambda_2 \mathcal{L}_{rec}(\hat{\mathcal{P}}, \mathcal{P})$

like architecture. Our skeletal-motion diffusion network progressively contracts and then expands (rebuilds) the spatial dimension of the input sequence of poses. To account for the temporal dimension of the input sequences, we build the U-Net with space-time separable GCN (STS-GCN) layers proposed in [164]. The conditioning autoencoder relies on STS-GCN to reconstruct the past motion and embed it into a latent space used to condition the diffusion.

In detail, the U-Net takes in input $X^{k+1:N}$ and a motion-temporal conditioning signal $h + \tau_\theta(t)$ which provides the network with the diffusion timestep and the encoded past-motion information. Furthermore, to align the dimensionality of this conditioning signal with that of the network's layers, the former is fed into an embedding layer projecting it to the correct vector space. This embedded conditioning signal is then fed to each STS-GCN layer. The contracting process of the U-Net progressively aggregates the joints of the poses (i.e., $\mathbb{R}^J \rightarrow \mathbb{R}^d$ s.t. $d \leq J$) while the expansion part "deconvolutes" the joints' vector space until it reaches the original space $\mathbb{R}^J$. Residual connections are present between specular layers.

4.3.5 MoCoDAD Algorithms

In this section, we outline the algorithms designed for both the training and inference phases of our proposed model (cf. Sec. 4.3.2). 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.

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 $\mathcal{G}$). 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

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

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

$\mathcal{H}_a$ 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:

$$\text{AS}[f_1 : f_N] = \text{mean}(\mathcal{S}) + \log \frac{1 + \max(\mathcal{S})}{1 + \min(\mathcal{S})}. \tag{4.10}$$

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:

$$\log\left(1 + \max\left(\mathcal{S}\right)\right) - \log\left(1 + \min\left(\mathcal{S}\right)\right). \tag{4.11}$$

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.

4.3.6 Implementation details

As in [51, 119, 127], 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. 4.6.4). We set $\lambda_1 = \lambda_2 = 1$. We train the network end-to-end with the Adam optimizer [91] 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 [130].

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 [178]. Our training took approximately 7 hours on an Nvidia Quadro P6000 GPU.

4.4 Experiments

Here we compare MoCoDAD with SoA approaches and provide a detailed discussion on the achieved performances.

As done in [2, 10, 51, 56, 108, 114, 119, 127], we report the *Receiver Operating Characteristic Area Under the Curve* (ROC-AUC) to assess the quality of MoCoDAD predictions on UBnormal [2], and the HR filtered [51] versions of STC [112], Avenue [110], and UBnormal.

4.4.1 Datasets

We use the UBnormal dataset [2] which contains 29 scenes synthesized from 2D natural images with the Cinema4D software. Each scene appears in 19 clips featuring both normal and abnormal events. The split into train, validation, and test adheres to the open-set policy, providing disjoint sets of types of anomalies for training, validation, and test; in accordance with the OCC setting, we only include normal actions for the training set. We also consider the processed poses and the human-related (HR) filtering of the dataset proposed by [51]. To compare with other state-of-the-art methods, we also experiment on the HR versions of the ShanghaiTech Campus (STC) [112] and the CUHK Avenue [110] datasets introduced by [127]. The former includes 13 scenes recorded with different cameras, for a total of about 300,000 frames, and 101 testing clips with 130 anomalous events. The latter consists of 16 training videos and 21 testing videos with a total of 47 anomalous events.

4.4.2 Comparison with state-of-the-art

Leading OCC techniques. We compare against SoA OCC techniques. Among these, MPED-RNN [127] combines the reconstruction and prediction errors of a two-branches-RNN to spot anomalies. Normal Graph [114] uses spatial-temporal GCNs (ST-GCN) to encode skeletal sequences. GEPC [119] encodes the input sequences with an ST-GCN, and clusters the embeddings in the latent space. Multi-timescale Prediction [151] encodes the observed input sequence and predicts future poses at different time scales through intermediate fully-connected layers. Both PoseCVAE [82] and BiPOCO [88] exploit a Conditioned Variational-Autoencoder to learn a posterior distribution of normal actions and use encoded past and future sequences to reconstruct the future one. The former uses an MLP-based architecture, while the latter is GRU-based. STGCAE-LSTM [103] reconstructs the past pose sequence and predicts the future one by an

Table 4.1: Comparison of MoCoDAD against SoA in terms of AUC on the three Human-Related datasets (i.e., HR-STC, HR-Avenue, and HR-UBnormal) and UBnormal. OCC skeleton-based techniques are marked with a $*$.

		HR-STC	HR-Avenue	HR-UBnormal	UBnormal
Conv-AE [67]	*CVPR '16*	69.8	84.8	-	-
Pred [108]	*CVPR '18*	72.7	86.2	-	-
MPED-RNN [127] $*$	*CVPR '19*	75.4	86.3	61.2	60.6
GEPC [119] $*$	*CVPR '20*	74.8	58.1	55.2	53.4
Multi-timescale Prediction [151] $*$	*WACV '20*	77.0	88.3	-	-
Normal Graph [114]	*Neurocomputing '21*	76.5	87.3	-	-
PoseCVAE [82] $*$	*ICPR '21*	75.7	87.8	-	-
BiPOCO [88] $*$	*Arxiv '22*	74.9	87.0	52.3	50.7
STGCAE-LSTM [103] $*$	*Neurocomputing '22*	77.2	86.3	-	-
SSMTL++ [10]	*CVIU '23*	-	-	-	62.1
COSKAD [51] $*$	*Arxiv '23*	77.1	87.8	65.5	65.0
MoCoDAD $*$		**77.6**	**89.0**	**68.4**	**68.3**

LSTM-based autoencoder. SSMTL++ [10] extends [56]; it replaces the convolutions with a transformer, changes the object detection backbone, and adds a few auxiliary proxy tasks. COSKAD [51] builds on STS-GCN to map the embeddings of normal poses into a narrow region in the latent space.

Results. In Table 4.1, we compare MoCoDAD and SoA methods on the three HR datasets and on UBnormal. MoCoDAD achieves the best AUC score of 77.6, 89.0, and 68.4 on HR-STC, HR-Avenue, and HR-UBnormal, respectively. Our proposed method outperforms the current best [51], up to 4.4%, demonstrating the importance of considering a range of different possible futures for each sample. Additionally, MoCoDAD achieves an AUC of 68.3 on the full UBnormal dataset, surpassing COSKAD by 5.1%. This can be due to the improved sensitivity that emerges from our proposed model: considering more than a single deterministic future smooths the prediction of MoCoDAD avoiding penalizing excessively hard-still-normal samples which would be considered anomalous based on the reconstruction error of a deterministic model.

Results VS. Supervised and Weakly Supervised methods. Table 4.2 evaluates MoCoDAD with supervised and weakly supervised methods reported in [2]. Notice that, despite the absence of supervision or visual information, MoCoDAD is competitive with methods exploiting a stronger form of supervision. In detail, MoCoDAD (68.3) outperforms the weakly supervised method [56] (59.3) and other fully supervised methods and is competitive with [16] (68.5). Further, our approach only presents a fraction of the parameters of its competitors. Notably, MoCoDAD is $\sim 852\times$ smaller than the current best [16].

Table 4.2: Comparison of MoCoDAD against supervised (†) and weakly supervised (‡) methods introduced in [2] in terms of AUC on the UBnormal dataset.

	Params	UBnormal
Sultani et al. [167] †	-	50.3
AED-SSMTL [56] †	>80M	61.3
TimeSformer [16] †	121M	**68.5**
AED-SSMTL [56] ‡	>80M	59.3
MoCoDAD	**142K**	68.3

Table 4.3: 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. [167] †	51.8
AED-SSMTL [56] †	68.2
TimeSformer [16] †	86.1
AED-SSMTL [56] ‡	58.5
MPED-RNN [127] $*$	61.2
GEPC [119] $*$	47.0
COSKAD [51] $*$	76.4
MoCoDAD $*$	**77.6**

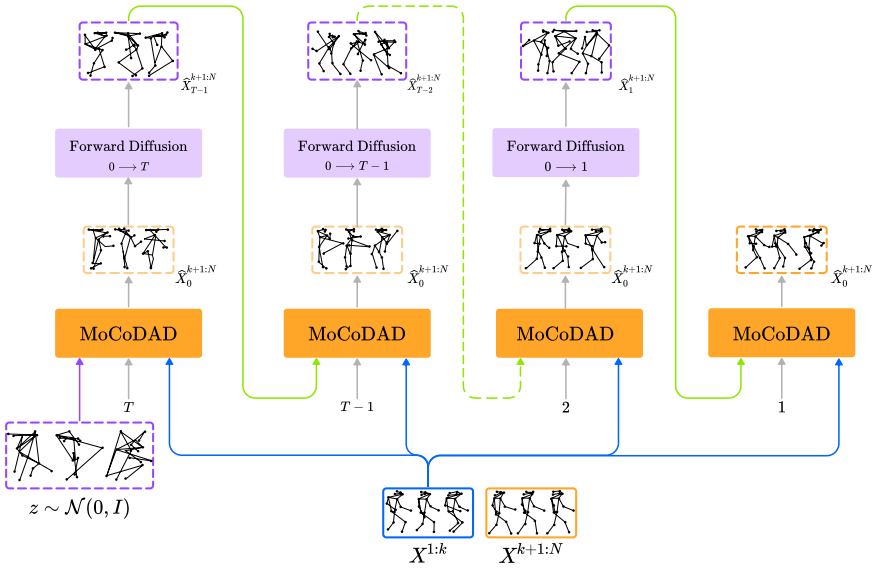


Figure 4.4: The iterative sampling process of our proposed method. At each step, MoCoDAD generates a prediction (light orange dashed boxes) employing a pose (purple dashed boxes) displaced proportionally to the current timestep t (when $t = T$ we just sample from random noise), together with a prior 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).

4.4.3 Results on the UBnormal Validation set

For completeness purposes, as done in [2], we report MoCoDAD's performances vs SoA on the validation set of UBnormal.

Table 4.3 shows that the validation set results align with those on the test sets reported in Tab. 4.1. 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 [16].

4.5 Qualitative Analysis

4.5.1 Generating motion sequences

This section visually illustrates how a sample is generated using the reverse procedure (Fig. 4.4). This supplements the discussion presented in Sec. 4.3.5, providing a visual explanation of Eq. 4.12. MoCoDAD generates motion sequences depending on a particular conditioning signal. This process is shown graphically in Fig. 4.4. Random noise x_T in the dimensions corresponding to the desired motion is initially sampled. The process then proceeds iteratively from step T to 1. MoCoDAD predicts a clean sample x_0 at each step t, then diffuses back to the previous X_{t-1}.

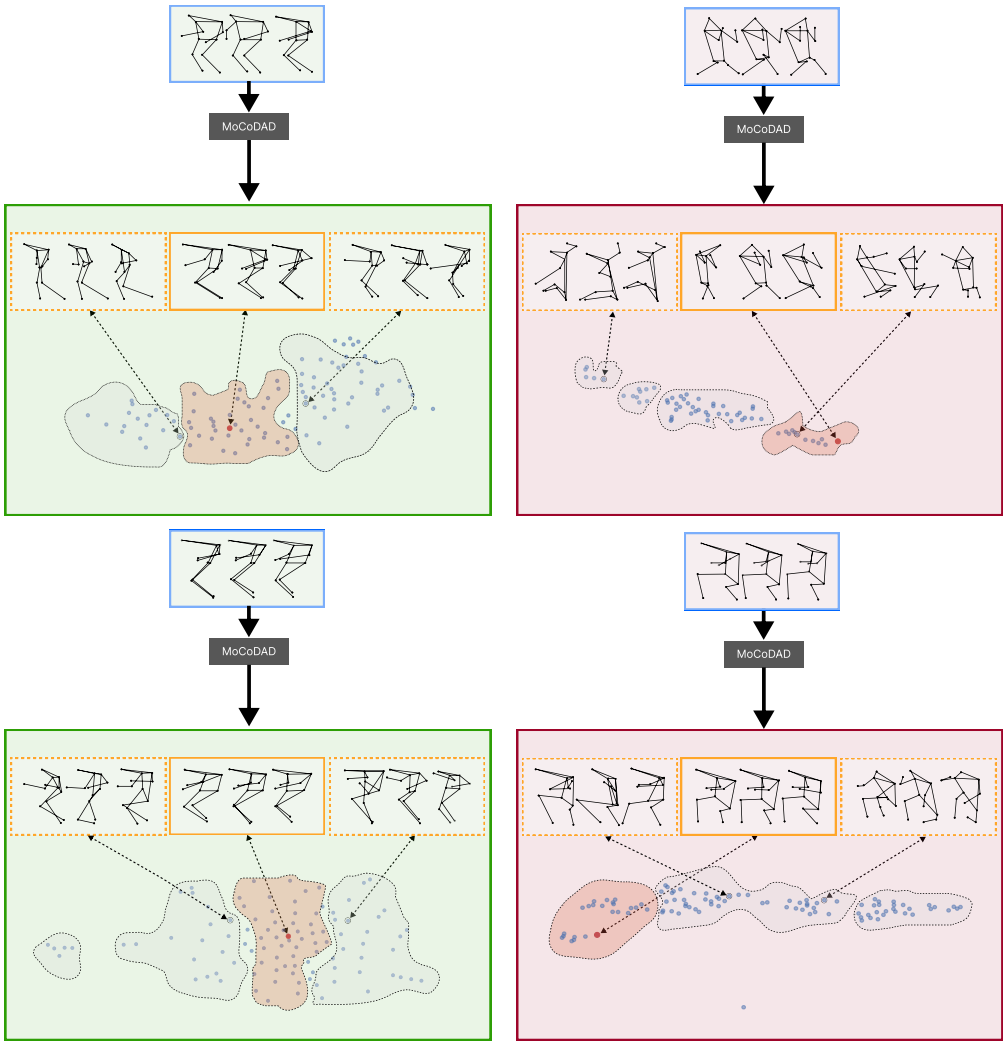


Figure 4.5: MoCoDAD detects anomalies by synthesizing and statistically aggregating multimodal future motions, conditioned on past frames. Red (right) and green (left) represent examples of anomaly and normality. At the bottom, 100 futures (2d mapped via t-SNE) are generated (dashed-orange rectangles) 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.

4.5.2 Qualitative results

Fig. 4.5 reveals that the generations produced with normal conditioning are biased towards the true future. The figure illustrates the t-SNE [177] 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

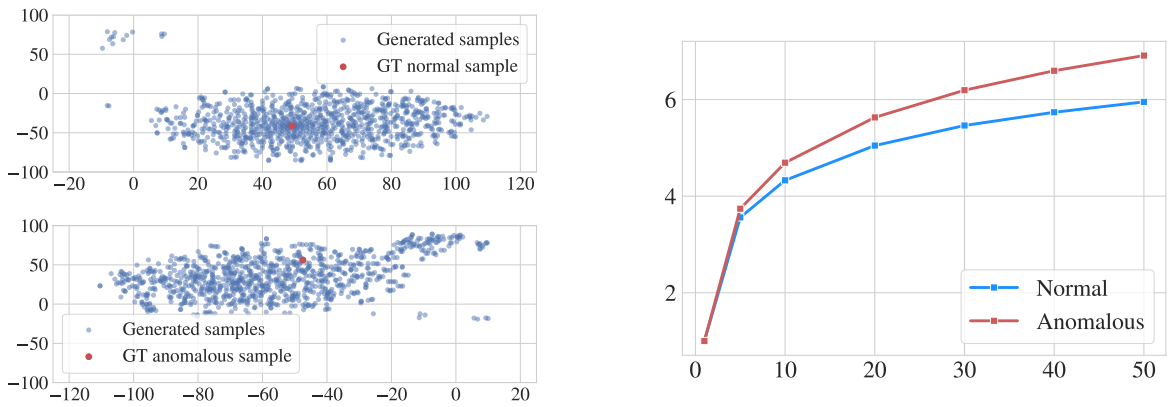


Figure 4.6: (*left*) Distribution of 1000 generated future motions, when conditioning on a normal past motion (top) and on an abnormal one (bottom). 2-dimensional projections are estimated via t-SNE [177]. Note how the true future motion (red dot) lies within a main distribution mode in the case of normality, but it lies in a marginal region for abnormality. (*right*) Plot of the diversity ratio rF [135], measuring the diversity of the generated future motions for normal and anomalous conditioning pasts. Moving along the x-axis, with more generated motion, the rF measures grow (MoCoDAD generates multimodal diverse motion) but they remain comparable (generating from normal and abnormal is anyhow multimodal).

the upper-right corner, and the second generation in the lower-right corner).

4.6 Ablation Studies

Here, we delve into a detailed discussion about the multimodal future generations (cf Sec. 4.6.1), the influence of the statistical aggregation of multiple generations and the normal Vs. anomalous conditioning (cf. Sec. 4.6.2), the effect of different conditioning strategies (cf. Sec. 4.3.3). Additionally, we discuss forecasting proxy tasks (Sec. 4.6.4) and analyze MoCoDAD's performances with the diffusion process applied to the latent space. (Sec. 4.6.5).

4.6.1 Multimodality

As clarified in Sec. 4.3.5, MoCoDAD engenders several outputs starting from a single conditioning motion by repeating the generative process m times. Thus, we question whether MoCoDAD can generate diverse multimodal motions and whether conditioning on normal or abnormal motions affects diversity. We provide an example of generated future motions in Fig. 4.6 (left), where we project the samples in 2D with t-SNE [177] for better visualization. Notice that both sets of generations have similar variance, showing that MoCoDAD produces diverse samples with both normal and abnormal conditioning sequences. Motions stemming from normal conditioning are biased toward the true future since they are mapped around it. Whereas, in the case of abnormal conditioning, the ground truth motion lies on the edge of the predictions' region, hence being correctly predicted with a lower chance.

We also measure diversity, employing the rF diversity metric from literature [26, 135]. We visualize the rF trend when increasing the number of generations in Fig. 4.6 (right). If the generated motion had no diversity, e.g., when generating only once, the rF would be equal to 1. Rather, the diversity ratio monotonically increases with the number of generations for both normal and abnormal cases, i.e., the more generated samples, the larger the empirical mode coverage and the diversity. We note that, following intuition, the diversity for abnormal cases grows slightly larger than for normal cases, but the rF measures

remain comparable.

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:

$$\text{Multimodality}(s) = \frac{1}{S_m} \sum_{i=1}^{S_m} \|\mathbf{a}_i - \mathbf{b}_i\|_2 \tag{4.12}$$

Compared with Fig. 4.8, the plot in Fig. 4.7 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.

4.6.2 Statistical aggregations of generated motions

We evaluate the anomaly detection performance when varying the number of generations m and the aggregation strategy for the anomaly score S (cf. Sec. 4.3.2). Fig. 4.8 (right) shows that the AUC positively correlates with the number of generated future motions for quantiles $Q < 0.5$, while the correlation is negative for the mean estimate and $Q > 0.5$. Such correlation can be better understood by looking at Fig. 4.8 (left), which depicts the average reconstruction error *probability density function* (PDF) for $m = 50$: when conditioned on normal past motions, MoCoDAD produces generations which are centered around the true future and, thus, is more likely to yield lower error scores than when generating with abnormal conditioning. The performance saturates for $m > 50$.

4.6.3 Conditioning

Here, we provide the reader with the performances of our method with different encoding approaches of the past (i.e., how the past motion is provided to the model) from which the future is generated (see Table 4.4).

As illustrated in Sec. 4.3.3, *Input Concatenation* concatenates the clean past frames to the corrupted part of each motion sample and feeds it directly to the denoising module. This strategy surpasses recent baselines (cf. Table 4.1) with AUC scores of 64.96 and 65.20 on UBnormal and HR-UBnormal, respectively. We deem this strategy suboptimal since it does not allow the past motion conditioning to be injected at each

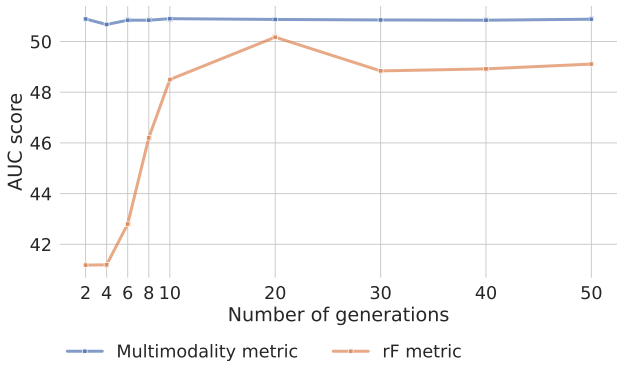


Figure 4.7: Anomaly detection performance trend when assuming a diversity metric as the anomaly score. It is worth noting that the rF metric yields results that are below the chance level.

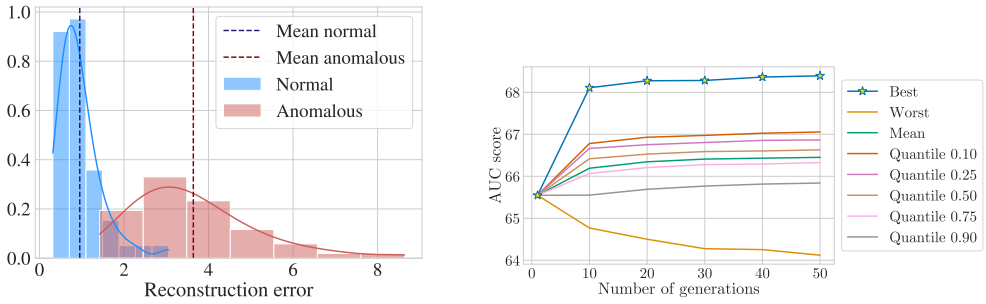


Figure 4.8: (*left*) Histograms of the reconstruction errors for 50 synthesized future motions, computed on the HR-UBnormal test set, for the case of conditioning on normal and abnormal past motions. (*right*) Correlation between the AUC scores and the number of generations, with each curve corresponding to a different aggregation statistic.

Table 4.4: Ablation study on the different methods for integrating conditioning information into the model.

	HR-UBnormal	**UBnormal**
Input Concatenation	65.2	65.0
E2E-embedding	64.4	64.2
AE-embedding (MoCoDAD)	**68.4**	**68.3**

layer (as with the embedding strategies), and, thus force the network to “remember" this information rather than focusing on the denoising of the future.

The *E2E-embedding* strategy encodes the motion history in the clean past frames and provides it to the latent layers of the denoising model, but it does not improve performance. We believe this happens because the learned embedding is not supervised, and it may not be representative enough of the original motion.

AE-embedding accounts for the best performances. It couples the encoder E with a symmetric decoder D and trains the whole model with an auxiliary loss $\mathcal{L}_{rec}$ which supervises the reconstruction of the first part of the motion (cf. Sec. 4.3.3). This past encoding strategy reaches 68.3 and 68.4 on UBnormal and HR-UBnormal, outperforming SoA techniques.

The first column of Table 4.7 refers to the timesteps used at inference time (γ) and the ones used at training (T). Following [188], 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 [138]. Fig. 4.9 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 is reflected in performance. Table 4.7 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

Table 4.5: Ablation study on the type of conditioning information to feed into the model to generate the missing frame.

	HR-UBnormal	UBnormal
Random Imputation	65.2	65.1
In-between Imputation	65.7	65.7
Forecasting (MoCoDAD)	**68.4**	**68.3**

true futures, and less similar to abnormal true futures. Note, however, that missing a condition on the past will result in general futures unrelated to the specific past, just normal. The results in Table 4.7 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.

4.6.4 Proxy Task

In Table 4.5 we assess the effectiveness of the forecasting proxy task (cf. Sec. 4.3.2). To this end, instead of limiting ourselves towards a rigid past-future split of poses, we explore two additional proxy tasks: i.e., in-between and random imputation. For in-between imputation, we corrupt the central $N - k$ poses and reserve the start and end of the sequence to condition the diffusion. For random imputation, we randomly select $N - k$ poses out of the full motion and corrupt them; the rest of the sequence is used for conditioning. Table 4.5 shows that random imputation is the worst proxy task (AUC of 65.10 and 65.21 for UBnormal and HR-UBnormal, respectively). Randomly splitting the sequence into two parts introduces inter-pose temporal gaps. Hence, we believe that this makes the reconstruction of non-contiguous motion more cumbersome. In-between imputation reaches an AUC of 65.65 and 65.72, respectively. We think that "stitching" the central motion to its endpoints is an easier task leading both normal and abnormal motions to be equally well reconstructed. Finally, forecasting is the best-performing proxy task with an AUC of 68.3 and 68.4 for UBnormal and HR-UBnormal.

4.6.5 Latent space

Diffusion on latent space has been adopted to improve computational efficiency in various domains [153]. MLD [191] is one of the most recent works in motion forecasting that applies the diffusion process directly in the latent space.

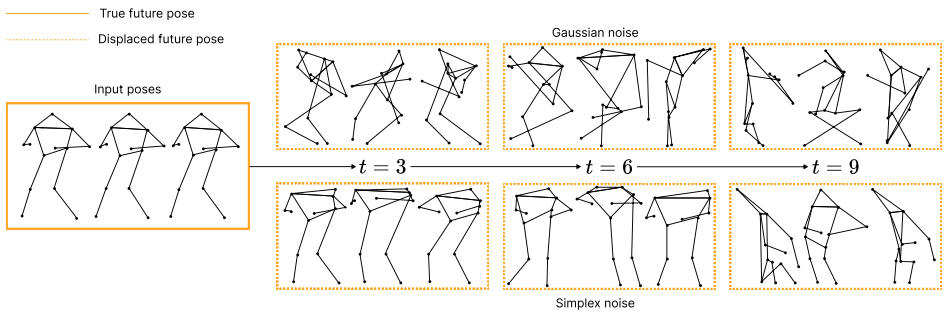


Figure 4.9: Comparison of Gaussian (up) vs Simplex (down) noises applied to a sequence of future poses.

Here, we propose an analysis that assesses the performance of MoCoDAD's diffusion process in the latent space. Thus, we propose two latent variations of MoCoDAD: MoCoDAD+MLD and Latent-MoCoDAD. For MoCoDAD+MLD we use a VAE to produce a latent representation $Z \in \mathbb{R}^d$ of the uncorrupted sequence $X^{k+1:N}$ and we perform the diffusion process on Z with the transformer-based denoising model as described in [191]. For Latent-MoCoDAD we rely on an STS-VAE to learn Z given $X^{k+1:N}$ and use the diffusion process, proposed in Sec. 4.3.2, on Z. For both these variants, we feed the corrupted Z to an MLP with conditioning signal $h + \tau_\theta(t)$ to reverse the diffusion and leave the conditioning component as is in the original MoCoDAD (i.e., encoding of history poses $X^{1:k}$).

Although the described variants can be trained end-to-end, we achieved the best performances by pre-training the autoencoder and training the diffusion module in the learned latent space keeping the autoencoder module frozen. We set the latent space dimension to $d = 16$ and the MLP blocks to $(16, 8, 16)$ interleaved with ReLU activations.

Table 4.6 illustrates the performances in terms of AUC-ROC of MoCoDAD and its latent variants on UBnormal and HR-UBnormal. Notice that the latent variants underperform w.r.t. the original MoCoDAD. We suspect that diffusion on latent spaces is not effective in skeleton-based VAD due to the lightweight encoding of the skeletons. This adheres with the literature since also MLD [191] shows that applying diffusion in the latent space underperforms w.r.t. the original proposal of the model.

4.6.6 Weaker forms of conditioning

In Table 4.7, we complement the previous section with an additional discussion on the forms of conditioning. Following the approach proposed by [188], 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.

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.

4.6.7 Diffusive steps

Sec. 4.3.2 delineates a gradual corruption technique that employs a displacement map in accordance 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 4.8, we consider five different

	HR-UBnormal	**UBnormal**
MoCoDAD+MLD	58.1	58.1
Latent-MoCoDAD	62.6	62.5
MoCoDAD	**68.4**	**68.3**

Table 4.6: AUC-ROC performance of diffusion on latent vs original space.

Table 4.7: 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**

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 [72, 130, 146, 172] 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 4.4. 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.

4.7 Discussion

We have presented a novel approach that models and exploits the diversity of normal and abnormal motions. In the former case, the forecast motions are multi-modal and pertinent to the observed portion of the sequence, as the model understands the underlying action. In the case of abnormal, the forecast motions do not show a bias towards the true future as the model does not expect any obvious outcome. Motivated by the improved mode coverage, this work has been the first to take advantage of probabilistic diffusion models for video anomaly detection, including a thorough analysis of the main design choices. MoCoDAD sets a

Table 4.8: 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

novel SoA among OCC techniques, and it catches up with supervised techniques while not using anomaly training labels.

Chapter 5

PREGO: online mistake detection in PRocedural EGOcentric videos

5.1 Overview

Egocentric procedure learning is gaining attention due to advancements in Robotics and Augmented Reality (AR) technologies. These technologies are pivotal in improving online* monitoring by providing timely feedback to the operator. Furthermore, these technologies are currently being adopted in a growing number of factories worldwide and are emerging as a critical tool for enhancing productivity, especially in the manufacturing industrial domain.

Recent works have produced numerous datasets [13, 44, 46, 57, 83, 123, 144, 159, 169, 169, 170, 182, 207], methodologies aimed at advancing procedure learning [57, 83, 111, 170, 183, 204] and error detection models [44, 169, 182]. Despite these advancements, as outlined in Table 5.1, prevalent methodologies typically focus on isolated procedural errors, like pinpointing missteps in sequence or identifying omitted steps within a procedure. They are unsuitable for situations requiring dynamic decision-making, specifically in an *online* context, or in cases where mistakes can be unexpected—hence, characterized as an *open-set*.

In this chapter, motivated by the previous studies described in Secs. 3 and 4, we propose the first model to detect PRocedural errors in EGOcentric videos (PREGO), which is both online, thus causal, and open-set, not limited to specific kinds of procedural mistakes. The online attribute is achieved by analyzing input videos sequentially up to a given frame t, ensuring that no future actions influence the current step recognition. On the other hand, the open-set learning is performed by exclusively exposing PREGO to correct procedural sequences when predicting mistakes, following the One-Class Classification (OCC) paradigm which proved effective in the sibling field of VAD in the previous chapters. Any step within a procedure that significantly diverges from the expected correct patterns is deemed an error, allowing PREGO to recognize a wide range of procedural mistakes without being confined to a restricted set of predefined errors.

PREGO's architecture is dual-branched, as depicted in Fig. 5.1. The first branch, the *step-recognition branch*, analyzes frames in a procedural video up to a current time t, aiming to classify the action being undertaken by the operator. This branch capitalizes on a state-of-the-art video-based online step recognition model, the OadTR [183] Encoder module. Concurrently, the second branch is in charge of *step-anticipation*:

*Most workflows can be aided by *online* monitoring algorithms, which provide feedback to the operator in due course. However, they may lag due to processing or connectivity delays. We distinguish online from real-time, whereby the second has strict requirements of instantaneous response.

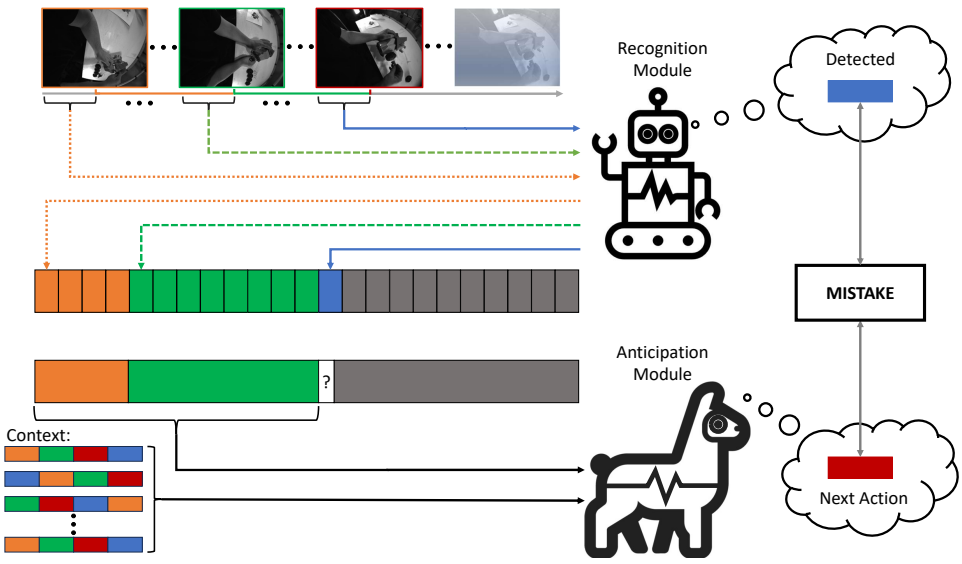


Figure 5.1: PREGO is based on two main components: The recognition module (top) processes the input video in an online fashion and predicts actions observed at each timestep; the anticipation module (bottom) reasons symbolically via a Large Language Model to predict the future action based on past action history and a short context of previous action sequences. Mistakes are identified when the currently detected action differs from the one anticipated from past action history (right).

a Large Language Model (LLM) [175] assesses the sequence of previously predicted actions up to time $t-1$, and predicts the t-th action in a zero-shot manner. An error is detected upon a misalignment between the currently recognized action and the anticipated one, thereby signaling a deviation from the expected procedure.

For the first time, we propose to perform next action prediction using a pre-trained Large Language Model (LLM) [175] for zero-shot symbolic reasoning through contextual analysis [64, 125, 168]. Utilizing correctly executed procedures as instances in the query prompt obviates the necessity for additional model fine-tuning and leverages the pattern-completion abilities of LLMs. Our proposed approach is an abstraction from the video content. Using labels allows for longer-term reasoning, as a label summarizes several frames. Also, this approach is an alternative to the carefully constructed action inter-dependency graphs [5]. We demonstrate that symbolic reasoning subsumes understanding long procedures and the action inter-dependencies, suggesting repositioning from semantic-based expressions of procedures to an implicit representation, where only patterns of symbols have to be recognized and predicted. By representing procedures as sequences and their steps as symbols, we let the predictor focus on the patterns that characterize the correct procedures.

To support the evaluation of PREGO, we review the procedural benchmarks of Assembly101 [159] and EpicTent [83], formalizing the novel task of online procedural mistake detection. In the reviewed online mistake detection benchmarks, which we dub Assembly101-O and EpicTent-O, the model is tasked with detecting when a procedural mistake is made, thus compromising the procedure. This occurs when performing an action makes it impossible to complete the procedure without a correction procedure. The compromising mistake may be a wrong action or a relevant action performed in such an order that the action

Table 5.1: Comparison among relevant models. In the modalities column, *C* stands for RGB images, *H* for hand poses, *E* for eye gaze, *K* for keystep labels, *D* for depth. Differently from previous works, we are the first to consider an egocentric one class and online approach to mistake detection.

	Ego	OCC	Online	Modalities	Task	Datasets
Ding et al. [44] - *Arxiv '23*				*K*	Mistake Detection	Assembly101 [159]
Wang et al. [182] - *ICCV '23*	✓			*C+H+E*	Mistake Detection	HoloAssist [182]
Ghoddoosian et al. [57] - *ICCV '23*				*C*	Unknown sequence detection	ATA [57] and CSV [143]
Schoonbeek et al. [169] - *WACV '24*	✓		✓	Multi	Procedure Step Recognition	IndustReal[169]
PREGO	✓	✓	✓	*C*	Mistake Detection	Assembly101 [159], Epic-Tent [83]

dependencies are not respected.

We summarize our contributions as follows:

- We present PREGO, the first method designed for online and open-set detection of procedural errors in egocentric videos. PREGO's online feature ensures causal analysis by sequentially processing input videos up to a given frame, preventing future actions from influencing current step recognition.
- PREGO achieves open-setness by exclusively relying on correct procedural sequences at training time, following the One-Class Classification (OCC) paradigm. This allows PREGO to identify a wide range of procedural mistakes, avoiding confinement to a predefined set of errors and avoiding the need for fine-grained mistake annotations.
- We propose using a pre-trained Large Language Model (LLM) for zero-shot symbolic reasoning through contextual analysis to predict the next action.
- To evaluate PREGO, we introduce the novel task of online procedural mistake detection and re-arrange existing dataset to provide two new benchmarks, referred to as Assembly101-O and EpicTent-O.

5.2 Related Work

5.2.1 Procedural Mistake Detection

Despite the recent growing attention to procedural learning [83, 159, 169, 182], the field lacks a unified approach to mistake detection, with fragmented literature and limited evaluations. The notable exception is [44], which uses knowledge graphs for error identification. It is worth noting that [44] method does not employ the video source, as it learns the procedures' steps directly from the transcripts of the procedures in [159]. Our proposal diverges from the aforementioned works as it leverages the video frames to detect the steps of the procedure and symbolic reasoning for an online assessment of the correctness of the procedure.

5.2.2 Steps recognition and anticipation

Step recognition is the task of identifying actions within a procedure. Indeed, a procedure is an ordered sequence of steps that bring to the completion of a task. Step recognition is pivotal in areas such as autonomous robotics and educational technology. Recent contributions in this domain include [161], which uses a novel loss for self-supervised learning and a clustering algorithm to identify key steps in unlabeled procedural videos. [111] introduces an action segmentation model using an attention-based structure with a Pairwise Ordering Consistency loss to learn the regular order of the steps in a procedure. Notably, they

devise a weakly supervised approach, using only the set of actions occurring in the procedure as labels, avoiding frame-level annotations. [205] approaches the task by leveraging online instructional videos to learn actions and sequences without manual annotations, blending step recognition with a deep probabilistic model to cater to the variability in step order and timing.

On the other hand, step anticipation focuses on predicting forthcoming actions in a sequence crucial for real-time AI decision-making. [1] addresses this by generating multiple potential natural language outcomes, pretraining on a text corpus to overcome the challenge of diverse future realizations. Additionally, the framework of [142] proposes solutions to future activity anticipation in egocentric videos, using contrastive loss to highlight novel information and a dynamic reweighing mechanism to focus on informative past content, thereby enhancing video representation for accurate future activity prediction. Unlike prior works, PREGO is the first model that anticipates actions via LLM symbolic reasoning in the label space.

5.2.3 Large Language Modelling and Symbolic Reasoning

Large language models are trained on large datasets and have many parameters. This gives them new capabilities compared to previous language models [185].
LLMs have shown excellent capabilities in modeling many natural language-related [175] and unrelated tasks [22, 65, 185]. We argue that there is a common ground between their next-token prediction mechanism and our action anticipation branch. Therefore, we incorporate LMs into our pipeline by resorting to a state-of-the-art model.

Recent research [49, 65, 105, 125, 133] has explored LLMs' ability to operate as *In-Context Learners* (ICLs), which means they can solve new tasks that they have never seen before. Given a query prompt with a context of input-output examples, LLMs can comprehend and address the problems in this setting without further fine-tuning.

LLMs as ICLs have been used for a variety of tasks, including planning [133], programming [65, 105], logical solvers [49], and symbolic reasoning [125].

Some work has shown that LLMs can continue semantically significant patterns [125], while other work has explored LLMs' in-context capabilities on semantically unrelated labels [186], where there is no relationship between a token and its meaning. Notably, recent research [131, 133] studied the opportunity to employ LLMs for devising plans to accomplish tasks. This is instrumental in robotics to enable automated entities to fulfill tasks. In our mistake detection pipeline, we leverage ICL using an LLM in the action anticipation branch. This LLM continues sequences of steps in a procedure, represented as symbols, given examples of similar procedures. Our LLM acts as a symbolic pattern machine, continuing the pattern of actions given a context of sequences performed in a goal-oriented way, even if the sequences do not follow a semantic scheme. This combines the challenges of predicting future actions and of having no semantics.

5.3 Methodology

Our framework for online mistake detection exploits a dual-branch architecture that integrates procedural step recognition with anticipation modeling, as depicted in Fig. 5.1. In the following sections, we elaborate on the problem formalization (Sec. 5.3.1), present the branches for step-recognition (Sec. 5.3.2) and step-anticipation (Sec. 5.3.3), and finally we illustrate the mistake detection procedure (Sec. 5.3.4).

5.3.1 Problem Formalization

We consider a finite set of N procedures $\{p_i\}_{i=1}^{N}$ that encodes the sequence of actions as $p_i = \{a_k\}_{k=1}^{K}$ where K varies depending on the specific procedure and $a_k \in \mathcal{A} = \{a | a \text{ is a possible action }\}$. Each procedure is also represented by a set of videos $\{v_i\}_{i=1}^{N}$ that, in turn, are composed of frames $v_i = \{f_\tau\}_{\tau=1}^{M_i}$ where M_i is the total number of frames in the video i.
Fixed a frame $f_{\bar{\tau}}$ from a given video $v_{\bar{i}}$, PREGO's task is double-folded: it has to (1) recognize the action $a_{\bar{\tau}}$ corresponding to the frame $f_{\bar{\tau}}$ in the video and (2) predict the action $a_{\bar{\tau}}$ that will take place at time $\bar{\tau}$ considering only past observations until time $\bar{\tau} - 1$.
The step recognition task is performed by a module ρ that takes as input the encoded frames of $v_{\bar{i}}$ up to $\bar{\tau}$ and returns an action $a_{\bar{\tau}}^{\rho}$. We then feed the module ξ, responsible for the anticipation task, with all the $a_1^{\rho}, ..., a_{\bar{\tau}-1}^{\rho}$ actions to have a prediction $a_{\bar{\tau}}^{\xi}$ for the next action in the obtained sequence.
Finally, we compare $a_{\bar{\tau}}^{\rho}$ with $a_{\bar{\tau}}^{\xi}$ and we deem as mistaken the actions where a misalignment between the outputs of the two branches occurs. For the sake of clarity, in the remainder of this section, we consider a single procedure p associated with a video v.

5.3.2 Step Recognition

We capitalize on the established OadTR [183] encoder that leverages the Transformers capabilities for online action detection to build our recognition branch. In fact, even if it is possible to make recognition using the encoder-decoder architecture, we show in Fig. 5.3 that the model benefits from having only the encoder module when coupled with a larger window W that includes more frames to produce its prediction. This choice leads to a better mAP considering different values for the window length and a drastic reduction of the parameters ($\sim 1/3$ compared to the original model).

In this setup, defining w as the size of the window W, the model predicts the action a_τ considering the frames $\{f_{\tau-w}, .., f_\tau\}$. This mechanism produces redundant results since the model eventually predicts the same action for consecutive frames. In order to obtain a consistent procedure, we simply consider unique actions every time the prediction for consecutive frames is the same.

5.3.3 Step Anticipation

We introduce a novel approach for step forecasting in procedural learning by harnessing the power of symbolic reasoning [125] through a Language Model (LM). Specifically, we employ a Large Language Model (LLM) for next-step prediction, feeding it with prompts constructed from procedural video transcripts. These prompts are structured in two parts: the first part comprises contextual transcripts CT extracted from similar procedures to inform the LLM about typical step sequences and order. The second part includes the current sequence of actions up to a specific frame, $f_{\bar{\tau}}$, detected by our module ρ, i.e.,

$$g_{\bar{\tau}} = [a_1^{\rho}, ..., a_{\bar{\tau}-1}^{\rho}] \tag{5.1}$$

This approach enables the LLM to utilize in-context learning, eliciting its ability to anticipate subsequent actions. Our framework operates in a zero-shot fashion, relying on the LLM's ability to retrieve the correct sequence continuation without specific training or fine-tuning but only leveraging the positive examples within the input prompts. Additionally, our method employs symbolic representations of the steps, converting the set of actions $\mathcal{A}$ into a symbolic alphabet Ω through an invertible mapping γ. Therefore, we can

express the symbolic predicted sequence as:

$$\gamma(g_{\bar{\tau}}) = [\gamma(a_1^\rho), ..., \gamma(a_{\bar{\tau}-1}^\rho)] = [\omega_1, ..., \omega_{\bar{\tau}-1}] \tag{5.2}$$

This conversion abstracts the actions from their semantic content, allowing the LLM to focus on pure symbols and sequences, thus simplifying the complexity of predicting the following action.
Finally, the ξ module, given the examples C and the current symbolic transcript $\gamma(g_{\bar{\tau}})$ described in its prompt, is required to output the most probable symbol $\omega_{\bar{\tau}}$ to continue the sequence. At this point, we apply the inverse function of γ to retrieve the underlying step label, i.e., $a_{\bar{\tau}}^\xi = \gamma^{-1}(\omega_{\bar{\tau}})$.

5.3.4 Mistake Detection

We finally compare the outputs of the two modules to detect procedural mistakes. Precisely, we consider as correct all the steps where the outputs of the two modules align with each other, while we deem as an error the cases for which the two outputs diverge. That is:

$$\begin{cases} a_{\bar{\tau}}^\rho \neq a_{\bar{\tau}}^\xi & \text{MISTAKE} \\ a_{\bar{\tau}}^\rho = a_{\bar{\tau}}^\xi & \text{CORRECT} \end{cases} \tag{5.3}$$

5.4 Benchmarking online open-set procedural mistakes

This section presents the benchmark datasets and the evaluation metrics used in our experiments. First, we introduce the reviewed online variants of Assembly101 and Epic-Tents (Sec. 5.4.1) and then we define the proposed online metrics in Sec. 5.4.2.

5.4.1 Datasets

We propose *Assembly101-O* and *Epic-Tents-O* as a refactoring of the original datasets [83, 159], detailing the selected labeling for online benchmarking, and the novel arrangement of training and test splits, to account for open-set procedural mistakes.

Assembly101-O

Assembly101[159] is a large-scale video dataset that enables the study of procedural video understanding. The dataset consists of 362 procedures of people performing assembly and disassembly tasks on 101 different types of toy vehicles. Each procedure is recorded from static (8) and egocentric (4) cameras and annotated with multiple levels of granularity, such as more than 100K coarse and 1M fine-grained action segments and 18M 3D hand poses. The dataset covers various challenges, such as action anticipation and segmentation, mistake detection, and 3D pose-based action recognition.
Assembly101 for online and open-set mistake detection (*Proposed*). We introduce a novel split of the dataset [159] that enables online, open-set mistake detection by design. Assembly101-O mainly encompasses two edits on [159], namely, a new train/test split and a revision of the length of the procedures. The novel split encloses all the correct procedures in the train set, leaving the videos with mistakes for the test

and validation set. This modification is needed to allow models to learn the sequences of steps that characterize correct procedures in a one-class classification fashion. In this way, models do not undergo the bias of learning specific kinds of mistakes during training; instead, as they are exposed exclusively to correct processes, they adhere to the One-Class-Classification (OCC) protocol and consider mistakes all the actions that diverge from the learned normalcy. As a further advantage, this saves all mistaken annotated videos for the test set, granting better balanced correct/mistaken validation and test sets and a more comprehensive evaluation of mistake detection. The second revision proposes considering each video for benchmarking until the procedure is compromised, i.e., until a mistake occurs due to mistaken action dependencies. Indeed, coherently with the OCC protocol, models are tasked with learning the correct flows of steps that allow procedures to be efficiently completed and considering sub-process after a mistake occurs creates a gap between the actions in the train set and those in the test, which prevents the models from recognizing or correctly anticipating the procedure steps. Moreover, this work proposes to focus on egocentric videos to be consistent with real-world applications. Hence, we only leverage a single egocentric video out of the eight views available for each video in [159].

Epic-Tent-O

Epic-Tent is a dataset of egocentric videos that capture the assembly of a camping tent outdoors. The dataset was collected from 24 participants who wore two head-mounted cameras (GoPro and SMI eye tracker) while performing the task. The dataset contains 5.4 hours of video recordings and provides annotations for the action labels, the task errors, the self-rated uncertainty, and the gaze position of the participants. The dataset also reflects the variability and complexity of the task, as the participants interacted with non-rigid objects (such as the tent, the guylines, the instructions, and the tent bag) and exhibited different levels of proficiency and uncertainty in completing the task.

Epic-Tent for online and open-set mistake detection (*Proposed*) This section introduces a novel split for the Epic-Tent dataset [83], designed to be adapted for the open-set mistake detection task. [83] is labeled with nine distinct mistake types. However, among these, "*slow*", "*search*", "*misuse*", "*motor*", and "*failure*" do not represent procedural errors, since, when they occur, the procedure is not tainted. On the other hand, the categories "*order*", "*omit*", "*correction*", and "*repeat*" are procedural mistakes. Our focus is solely on procedural mistakes for the specific task of mistake detection. Epic-Tent is designed for the supervised error detection task and, differently from [159], every reported procedure reports some mistakes, hampering the reproduction of the split procedure proposed for Assembly101-O. Nonetheless, this dataset provides the confidence scores assigned to each frame by the performer, indicating their self-assessed uncertainty during the task. Thus, we define a strategy for splitting, in which videos featuring the most confident performers form the train set, while those showing higher uncertainty (and thus potentially more prone to errors) populate the test set. This partitioning strategy holds encouraging promise, especially in real-world scenarios where the accurate labeling of erroneous frames is hard to achieve or where the training of a mistake detector can initiate immediately post-recording without necessitating the completion of the entire annotation process. Ultimately, the resulting split comprises 14 videos for the training set and 15 for the test set.

The Epic-Tent dataset showcases only egocentric videos recorded through Go-Pro cameras. This further highlights the practicality and relevance of the proposed novel benchmark in open-scene contexts. The videos in the test set are also trimmed up to the last frame of the first mistake occurring in the video, while those representing correct procedures are maintained unaltered.

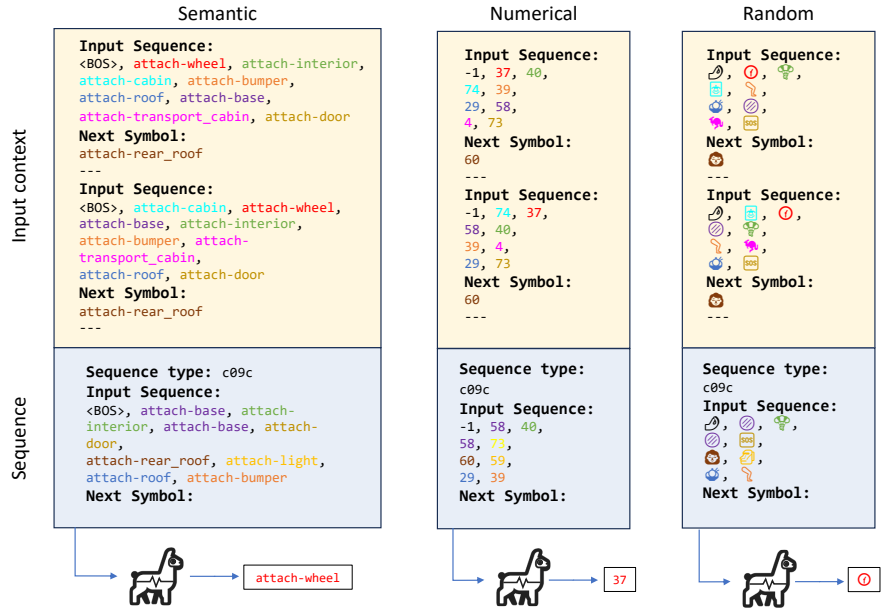


Figure 5.2: Three different representations of the actions in the prompt for the LLM model. On the left, the prompt is represented using action labels. In the center, the prompt is represented by label indices. On the right, the prompt is represented by random symbols.

5.4.2 Metrics

To assess the performance of our procedural mistake detection model, we use True Positives as a measure of the model correctly identifying errors and True Negatives as a measure of accurately labeling procedures that are not errors. Thus, we rely on the key Precision, Recall, and F1 score metrics to evaluate the performance of our model. These metrics offer valuable insights into the model's capability to identify and classify mistakes within procedural sequences. More specifically, precision quantifies the accuracy when predicting mistakes, minimizing false positives. Recall assesses the model's capability to retrieve all mistakes, reducing the number of false negatives. Finally, the F1 score is the harmonic mean of precision and recall, and it balances failures due to missing mistakes and reporting false alarms.

5.5 Experiments

In this section, we present the results of our experiments on online and open-set mistake detection in procedural videos. We contrast PREGO with several baselines that employ different mistake detector techniques or use part of the ground truth as an oracle.

All the baselines are assessed on the Assembly101-O and Epic-Tent-O datasets, detailed in section 5.4.1. The metrics used are F1 score, precision and recall, explained in 5.4.2. We first describe the baselines in section 5.5.1, and then analyze the results in 5.5.2, shown in Table 5.2.

5.5.1 Baselines

To estimate the effectiveness of PREGO, we evaluate its performance by comparing it against the following baseline models based on the metrics presented in 5.4.2:

One-step memory We define a *transition matrix* considering only the correct procedures. Specifically, given the set of the actions $\mathcal{A}$ in the training set with $|\mathcal{A}| = C$, we define a transition matrix $M \in \mathbb{R}^{C \times C}$

Table 5.2: Comparative evaluation of PREGO Vs the selected baselines on the task of procedural mistake detection on the Assembly101-O and Epic-Tent-O datasets.

	Assembly101-O			Epic-Tent-O		
	F1 score	**Precision**	**Recall**	**F1 score**	**Precision**	**Recall**
One-step memory	21.3	16.3	30.7	10.6	6.6	26.6
OadTR for MD[183]	20.7	24.3	18.1	10.2	6.7	21.7
BERT [42]	31.8	78.2	20.0	10.4	75.0	5.6
PREGO$_{(LLAMA)}$	33.4	20.4	91.7	13.8	7.54	80.0
PREGO$_{(DaVinci)}$	35.8	22.1	94.2	-	-	-
PREGO$_{(LLAMA)}$ - oracle Recognition	42.0	26.9	95.5	20.6	11.45	100
PREGO$_{(DaVinci)}$ - oracle Recognition	46.3	30.6	94.3	-	-	-

which stores in position (l, m) the occurences that action m follows action l. We then label as *mistake* the actions occurring in the test split that do not correspond to transitions recorded in the training set.

OadTR for mistake detection [183] proposes a framework for online action detection called OadTR that employs Transformers to capture the temporal structure and context of the video clips. The framework consists of an encoder-decoder architecture. The encoder takes the historical observations as input and outputs a task token representing the current action. The decoder takes the encoder output and the anticipated future clips as input and outputs a refined task token incorporating the future context. A discrepancy between the outputs of the encoder and the decoder produces a mistake

BERT [42] We leverage the capability of BERT utilizing its specific [CLS] token to predict the correct or erroneous sequence of action. More specifically, we fine-tune BERT using the next-sentence-prediction task, where the model is trained to predict whether one sentence logically follows another within a given text. In our context, we apply this to determine whether step B can follow another step A within a procedure. Here, steps are defined as sets of two words, such as *attach wheel*, representing coarse actions. To execute this, BERT is presented with pairs of sentences corresponding to procedures A and B, tasking it with predicting the sequential relationship between them. BERT's advantage lies in pre-training on a vast text corpus, followed by fine-tuning for our specific scenario. This process enables BERT to grasp contextual connections between sentences, rendering it effective for tasks like classifying procedures and comprehending the logical flow of information in text.

5.5.2 Results

We experimentally evaluate PREGO against the selected baselines on Assembly101-O and EpicTent-O. Results are reported in Table5.2

Considering Assembly101-O, the One-step memory approach achieves results similar to OadTR in terms of F1-score. However, both techniques are limited in the length of their temporal dependencies. In fact, OadTR is based on processing frames, and its performance flattens when surpassing 512 frames (cf. Fig. 5.3), only accounting for 64 frames, while procedures last in general 7 minutes. One-step memory reasons on the abstracted label space, where each label summarizes tens of frames. Yet, it only considers the single previous action, which limits its performance.

BERT demonstrates the advantages of incorporating LLM in the anticipation branch: by moving the reasoning to a higher level of abstraction, it shows more awareness of the past. In this way, BERT can accurately detect the mistakes by reducing the false alarms. However, the Recall highlights that many mistakes are not detected, revealing a conservative behavior. On the other hand, PREGO maximizes the level of abstraction by leveraging Symbolic Reasoning, enabling it to fully exploit the context information.

In this way, it can achieve the highest F1 score demonstrating that symbolic reasoning is a valuable strategy to detect mistakes. Between the two symbolic baselines, namely PREGO$_{Llama}$ and PREGO$_{DaVinci}$, the latter is more capable of producing abstract representation and emerges as the best performer.

Similar considerations apply to Epic-Tent-O with some variations on the scale of the metrics. We attribute these discrepancies to the differences between the two datasets. Indeed, there exist multiple valid procedures for assembling a tent, each dependent on different combinations and orders of actions, whereas industrial assembly processes enforce a stricter among steps as in the case of Assembly101-O. We do not test DaVinci for Epic-Tent-O because the processing service is provided at a cost that scales with the number of tokens in input and output, and this is much larger for Epic-Tent-O. By contrast, LLAMA is open-source, allowing for experimenting on it at scale.

5.6 Ablation Study

Here, we present the results on step anticipation using two different Large Language Models (LLMs) (c.f. Sec. 5.6.2) and compare them with the case of oracular step recognition (c.f. Sec. 5.6.2). We also investigate the effect of different semantic prompts on the performance of the LLMs (see Section 5.6.3).

5.6.1 Step Recognition

An interesting element to examine is the effect that the window size W has on the performance of the step recognition model. The window size refers to the number of previous frames that the model utilizes as input to predict the current action. We vary the window size from 64 to 512 and we measure the mean Average Precision (mAP) of the model on the Assembly101 dataset. The mAP evaluates the accuracy and completeness of the model in recognizing the actions. Fig. 5.3 reports the experimental results, showing that the mAP of the model increases as the history size increases. This indicates that the model gains from having more past context from which the model can learn a richer temporal structure of the procedures. In addition, reducing the architecture to the encoder only reduces the number of parameters by approximately 66%. While larger window sizes might enhance performance, they do not align with the trade-off between online action recognition and window size that allows for the initial prediction to begin at frame f_W.

5.6.2 Step Anticipation

Llama Vs DaVinci In this section, we evaluate the performance of different LLMs for procedural mistake detection with symbolic inputs. We follow the Step Anticipation setting described in Sec. 5.3.3 and compare the LLAMA-2 model [175] with the OpenAI DaVinci method. We report the results in Table 5.2, using F1 score, precision, and recall as the evaluation metrics. The table shows that the OpenAI DaVinci method achieves the highest F1 score on the Assembly101 dataset. We conclude that the OpenAI DaVinci method is the most effective LLM for procedural mistake detection, as it can learn the normal patterns of the procedures and detect deviations from them. We choose LLAMA as the LLM for the experiments on Epic-Tent and for the ablation studies due to its open source and free nature and its performance that is almost on par with DaVinci.

Oracular Evaluation To evaluate the performance of the action anticipation model's performance, we replace the step recognition branch's predictions with the ground truth action labels. This experiment mimics

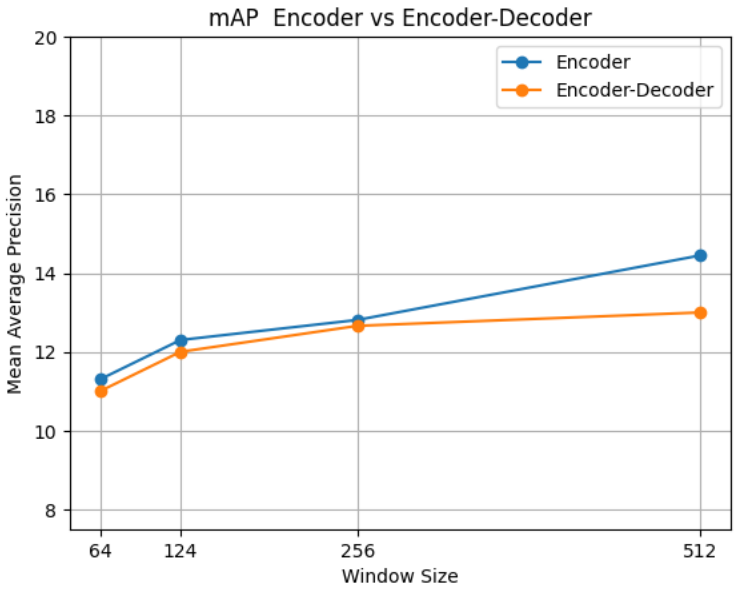


Figure 5.3: Relation between window size and mAP for the OadTR [183] Encoder Vs Encoder-Decoder architectures. Using only the Encoder with a bigger window size leads to better performance while saving on all the Decoder parameters

a situation where the video branch can perfectly recognize the actions performed in the videos. We refer to this experiment as “PREGO - oracle Recognition" in Table 5.2, comparing it with PREGO and the other baseline methods. As expected, the oracle recognition experiment outperforms the PREGO model, achieving an F1 score of 46.3 and 44.1 with LLAMA and DaVinci, respectively, compared to 35.8 and 31.2 of PREGO, and indicating that the accuracy of the video branch is a bottleneck for the overall performance. However, the oracle recognition experiment also reveals our model’s potential for improvement. Other factors influence the model’s performance, such as the quality of the symbolic inputs, the semantic prompts, and the LLM architecture.

5.6.3 Performance of different prompt types.

We investigate the effect of different action representations in the prompt for the Step Anticipation prediction. Following [125], we consider three ways of representing an action: numerical, semantic, or random symbols. Numerical representation means that an action label is replaced with an index in the range $[0, \mathcal{A}]$, where $\mathcal{A}$ is the total number of actions. Semantic representation implies that the action is represented by its action label. Random symbol indicates that each action is assigned to a different symbol, such as a set of emojis. This allows us to examine how the LLM can manage different levels of abstraction and expressiveness of the input prompt. Fig. 5.2 illustrates an example of a prompt in the three different representations.

Table 5.3 shows the experiment results using the described representations. We observe that all the different representations achieve close performance, with the numerical representation achieving the highest F1 score, with 33.4, followed by the random representation, 33.2. We hypothesize that the numerical representation is easier for the model to understand and predict the next step, as it reduces the number of tokens to be generated. Indeed, an action index requires only four tokens, whereas the emojis and the action labels require at least eight tokens. Remarkably, the semantic representation achieves a comparable performance even though words can introduce bias or ambiguity into the model. This indicates that the model can handle the natural language input and extract the relevant information for the step anticipation task. Surprisingly,

Table 5.3: Performance of PREGO with different prompt representations for Procedural Mistake Detection evaluated via F1 score, precision and recall on the Assembly101 dataset.

	F1 score	Precision	Recall
Random	33.2	19.9	98.7
Semantic	33.1	20.2	92.3
Numerical	33.4	20.4	91.7

the random symbol representation has a comparable performance to the other representations, even though the model does not have any semantic or numerical association with them. This suggests that the model can learn the temporal structure of the actions from the input history, regardless of the symbol representation.

5.7 Discussion

We have introduced PREGO, a one-class, online approach for mistake detection in procedural egocentric video. PREGO predicts mistakes by comparing the current action predicted by an online step recognition model with the next action, anticipated through symbolic reasoning performed via LLMs. To evaluate PREGO, we adapt two datasets of procedural egocentric videos for the proposed task, thus defining the ASSEMBLY101-O and Epic-tent-O datasets. Comparisons against different baselines show that the feasibility of the proposed approach to one-class online mistake detection. We hope that our investigation and the proposed benchmark and model will support future research in this field.

Chapter 6

Conclusions

Amidst the rapid advancements in AI and CV that promise to redefine industrial automation, this thesis directly confronts the challenges of ensuring utmost safety and enhancing operational efficiency in collaborative human-robot environments. While these innovations significantly improve production lines, they also bring to light critical safety and workflow management issues. Addressing these concerns head-on, the research develops innovative solutions to promote secure interactions and streamlined processes between humans and robots.

Indeed, our research has harnessed the latest in technological innovation, applying it within new contexts to significantly bolster both safety and operational efficiency, with a particular emphasis on enhancing processing speed towards achieving real-time execution. To this end, a key development has been the introduction of a pioneering graph convolutional network technique meticulously designed for precise and rapid human pose estimation. This advancement greatly enhances safety protocols in human-robot interactions, empowering cobots with the ability to quickly understand and respond to human actions, thus minimizing accident risks and reinforcing the reliability of industrial processes.

Building on the foundational advances in safety and efficiency, this research has also made significant strides in enhancing mistake detection within industrial procedural sequences. This field within Computer Vision, while not as extensively charted, presents unique challenges and opportunities. Recognizing procedural mistakes as phenomena with open-set characteristics, we developed a novel framework for Online Mistake Detection in egocentric procedural videos. Insights from the VAD domain deeply inform our approach; by understanding how VAD models identify unexpected events without prior specific training on them, we adapted similar principles to the dynamic and unpredictable nature of procedural mistakes, paving the way for a method that is both promising and applicable in real-world settings. This strategic leveraging of VAD concepts allows us to set a solid foundation for detecting a wide array of procedural errors, underscoring our commitment to operational safety and accuracy.

This research extends into the domain of anomaly detection with a focus on One-Class Classification methods and the exploration of behavior's multimodal nature, leading to the development of state-of-the-art methodologies for human-related VAD. Extensively validated, these methodologies serve the dual purpose of enhancing the academic domain and forging pathways toward practical implementations in mistake detection. The study of VAD, while not directly applied in industrial settings, illuminates the intricate dynamics between anomalies and procedural errors, laying a theoretical foundation that is instrumental for future endeavors in Online Mistake Detection and beyond, potentially influencing a range of sectors with its applicability.

Through the analysis of established literature and the development of new methodologies, this research underscores a commitment to refining the safety and efficiency of human-robot interactions. The methodologies and frameworks developed herein, while capitalizing on cutting-edge technological insights, offer pragmatic strategies to mitigate risks and aim at optimizing industrial operations.

The inquiry into VAD and the innovative exploration of Online Mistake Detection showcase the thesis's broader impact beyond immediate industrial applications. This research contributes to a growing body of knowledge that seeks to balance technological innovation with human-centric considerations, advocating for systems that are both advanced and accessible.

In summary, the contributions of this thesis extend beyond the confines of academic research, offering tangible strategies for the real-world challenges faced in industrial automation. Bridging the gap between theoretical exploration and practical application, it marks an important advancement in the ongoing dialogue on AI's role in shaping the industry's future. This work, therefore, stands as a testament to the evolving relationship between humans and robots, charting a course for future investigations that are as informed by technological potential as they are by the imperatives of safety and efficiency.

The research presented in this thesis lays a foundational framework for enhancing safety and efficiency in industrial human-robot collaboration through cutting-edge methodologies. While the outcomes have been promising, several directions for future work have emerged, aiming to further the field.

- **Optimization of Pose Estimation Models.** Further research should refine the proposal of pose forecasting models for HRC. While the proposed SeS-GCN reports encouraging results in terms of efficiency, to effectively achieve safety in spaces shared simultaneously by robots and humans, there is a need for models tailored for the industrial use case, capable of reacting in real-time. Real-time pose estimation is crucial to anticipate and prevent potential collisions, ensuring a safe interaction between robots and human workers. Without the ability to rapidly interpret and respond to human movements, the system may fail to react promptly to unexpected actions, compromising safety. Investigating methods to reduce computational demands while maintaining or improving accuracy could make these models more accessible for real-time applications across diverse industrial scenarios. Additionally, exploring advanced algorithms and hardware acceleration techniques could further enhance the responsiveness and reliability of pose estimation systems, ultimately fostering safer and more efficient human-robot collaboration.

- **Extending industrial benchmarks.** This work introduces three datasets specifically tailored for industrial contexts. On the other hand, we limit our focus to specific industrial contexts such as assembly lines. Future work should fill this gap, incorporating data from different sectors and more complex interactions between humans and robots, providing a richer resource for training and testing AI models. Expanding the dataset to include a wider range of industrial activities, such as logistics, quality control, and maintenance, would enhance the versatility and applicability of the pose forecasting models. Furthermore, capturing data in varied environmental conditions and from different camera angles can contribute to the robustness of the models. Ensuring the inclusion of movement patterns can further improve the generalizability of the models, making them more effective across various real-world scenarios. Broadening the range of data resources, including more general procedural activities, will ultimately lead to the development of AI models better equipped to handle the complexities of human-robot interaction in dynamic industrial environments.

- **Broaden the context of Mistake Detection.** The novel framework for Online Mistake Detection

developed in this thesis offers a promising direction for ensuring procedural integrity. Future work could focus on adapting and testing this framework in diverse industrial contexts beyond the initial settings considered here to assess its versatility and effectiveness in a broader range of procedures.

- **Exploring Further Applications of DDPMs and LLMs.** The use of diffusion and Large Language Models for anomaly detection and mistake prediction represents a novel approach in the field. Nevertheless, the field needs further insights, and future studies could explore other potential applications of these technologies in industrial settings, such as predictive maintenance, quality control, and operational planning. In predictive maintenance, these models could analyze patterns in equipment data to anticipate failures before they occur, thus reducing downtime and maintenance costs. For quality control, diffusion models could detect deviations from product specifications in real time, enhancing production accuracy and reducing waste. These technologies can forecast demand, optimize resource allocation, and improve overall efficiency in operational planning. Furthermore, investigating the integration of these models with existing industrial IoT systems could provide a seamless and scalable solution for real-time monitoring and decision-making. Developing user-friendly interfaces and visualization tools will also be crucial to ensure that the insights generated by these advanced models are accessible and actionable for industry professionals. By expanding the scope of application and refining these technologies, future research can significantly enhance their impact on industrial operations, driving innovation and improving productivity.

- **Robustness to Environmental Variability.** Ensuring that AI models are robust to variations in the industrial environment, such as lighting changes, noise, and occlusions, is essential for their reliability. Investigating techniques for enhancing model robustness and generalization across different settings and conditions would be a valuable direction for future work. Investigating techniques for enhancing model robustness and generalization across different settings and conditions would be a valuable direction for future work. This includes exploring advanced data augmentation methods, such as simulating various environmental conditions during training to improve the model's adaptability. Additionally, integrating sensor fusion techniques, where data from multiple sensors (e.g., cameras, LiDAR, thermal sensors) are combined, can enhance the model's ability to maintain performance despite challenging conditions. Another promising area of research is the development of self-learning systems that can continuously update and refine their algorithms based on new data and environmental changes. These approaches can significantly contribute to the creation of AI models that are not only robust and reliable but also capable of operating effectively in a wide range of industrial scenarios.

Continuing this journey of exploration and innovation, future endeavors based on these suggestions have the potential to advance industrial automation significantly. Building on the groundwork laid by this thesis, such research could lead to breakthroughs that transform industrial operations, making them safer, more efficient, and ready for the challenges of tomorrow.

Bibliography

[1] M. A. Abdelsalam, S. B. Rangrej, I. Hadji, N. Dvornik, K. G. Derpanis, and A. Fazly. Gepsan: Generative procedure step anticipation in cooking videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2988–2997, October 2023.

[2] A. Acsintoae, A. Florescu, M.-I. Georgescu, T. Mare, P. Sumedrea, R. T. Ionescu, F. S. Khan, and M. Shah. Ubnormal: New benchmark for supervised open-set video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20143–20153, 2022.

[3] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021.

[4] E. Aksan, M. Kaufmann, and O. Hilliges. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[5] K. Ashutosh, S. K. Ramakrishnan, T. Afouras, and K. Grauman. Video-mined task graphs for keystep recognition in instructional videos. *arXiv preprint arXiv:2307.08763*, 2023.

[6] D. J. Atha and M. R. Jahanshahi. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring*, 17(5):1110–1128, 2018.

[7] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv*, abs/1803.01271, 2018.

[8] M. Balcilar, G. Renton, P. Héroux, B. Gaüzère, S. Adam, and P. Honeine. Spectral-designed depthwise separable graph neural networks. In *Proceedings of Thirty-seventh International Conference on Machine Learning (ICML 2020)-Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

[9] A. Barbalau, R. T. Ionescu, M.-I. Georgescu, J. Dueholm, B. Ramachandra, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah. Ssmtl++: Revisiting self-supervised multi-task learning for video anomaly detection. *Computer Vision and Image Understanding*, 229:103656, 2023.

[10] A. Barbalau, R. T. Ionescu, M.-I. Georgescu, J. Dueholm, B. Ramachandra, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah. Ssmtl++: Revisiting self-supervised multi-task learning for video anomaly detection. *Computer Vision and Image Understanding*, 229:103656, 2023.

[11] A. Bauer, D. Wollherr, and M. Buss. Human–robot collaboration: a survey. *International Journal of Humanoid Robotics*, 5(01):47–66, 2008.

[12] E. P. Beltran, A. A. S. Diwa, B. T. B. Gales, C. E. Perez, C. A. A. Saguisag, and K. K. D. Serrano. Fuzzy logic-based risk estimation for safe collaborative robots. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, pages 1–5, 2018.

[13] Y. Ben-Shabat, X. Yu, F. Saleh, D. Campbell, C. Rodriguez-Opazo, H. Li, and S. Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021.

[14] K. Benesova, A. Svec, and M. Suppa. Cost-effective deployment of bert models in serverless environment, 2021.

[15] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

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

[17] E. Bertino, M. Kantarcioglu, C. G. Akcora, S. Samtani, S. Mittal, and M. Gupta. Ai for security and security for ai. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pages 333–334, 2021.

[18] S. E. Bibri, J. Krogstie, A. Kaboli, and A. Alahi. Smarter eco-cities and their leading-edge artificial intelligence of things solutions for environmental sustainability: A comprehensive systematic review. *Environmental Science and Ecotechnology*, 19:100330, 2024.

[19] D. Bogdoll, M. Nitsche, and J. M. Zöllner. Anomaly detection in autonomous driving: A survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4488–4499, 2022.

[20] R. Bogue. Strong prospects for robots in retail. *Industrial Robot: the international journal of robotics research and application*, 46(3):326–331, 2019.

[21] S. Bragança, E. Costa, I. Castellucci, and P. M. Arezes. A brief overview of the use of collaborative robots in industry 4.0: Human role and safety. *Occupational and environmental safety and health*, pages 641–650, 2019.

[22] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[23] J. Bütepage, H. Kjellström, and D. Kragic. Anticipating many futures: Online human motion prediction and synthesis for human-robot collaboration. *ArXiv*, abs/1702.08212, 2017.

[24] Y. Cai, L. Huang, Y. Wang, T.-J. Cham, J. Cai, J. Yuan, J. Liu, X. Yang, Y. Zhu, X. Shen, D. Liu, J. Liu, and N. M. Thalmann. Learning progressive joint propagation for human motion prediction. In *The European Conference on Computer Vision (ECCV)*, 2020.

[25] S. Calderara, U. Heinemann, A. Prati, R. Cucchiara, and N. Tishby. Detecting anomalies in people's trajectories using spectral graph analysis. *Computer Vision and Image Understanding*, 115(8):1099–1111, 2011.

[26] L. Calem, H. Ben-Younes, P. Pérez, and N. Thome. Diverse probabilistic trajectory forecasting with admissibility constraints. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3478–3484, 2022.

[27] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[28] A. Castro, F. Silva, and V. Santos. Trends of human-robot collaboration in industry contexts: Handover, learning, and metrics. *Sensors*, 21(12):4113, 2021.

[29] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[30] Y. Chang, Z. Tu, W. Xie, and J. Yuan. Clustering driven deep autoencoder for video anomaly detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 329–345. Springer, 2020.

[31] S. Chaudhary, S. Sharma, S. Mongia, and K. Tripathi. Artificial intelligence (ai) in healthcare: Issues, applications, and future. In *Concepts of Artificial Intelligence and its Application in Modern Healthcare Systems*, pages 1–17. CRC Press, 2024.

[32] J. H. Chen and K. T. Song. Collision-Free Motion Planning for Human-Robot Collaborative Safety under Cartesian Constraint. *IEEE Int. Conf. Robot. Autom.*, pages 4348–4354, 2018.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations, 2020.

[34] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

[35] Y. S. Chong and Y. H. Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *Advances in Neural Networks-ISNN 2017: 14th International Symposium, ISNN 2017, Sapporo, Hakodate, and Muroran, Hokkaido, Japan, June 21–26, 2017, Proceedings, Part II 14*, pages 189–196. Springer, 2017.

[36] M. Costanzo, G. De Maria, G. Lettera, and C. Natale. A multimodal approach to human safety in collaborative robotic workcells. *IEEE Transactions on Automation Science and Engineering*, PP:1–15, 01 2021.

[37] Q. Cui, H. Sun, and F. Yang. Learning dynamic relationships for 3d human motion prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6518–6526, 2020.

[38] M. Dallel, V. Havard, D. Baudry, and X. Savatier. Inhard - industrial human action recognition dataset in the context of industrial collaborative robotics. In *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, 2020.

[39] L. Dang, Y. Nie, C. Long, Q. Zhang, and G. Li. MSR-GCN: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[40] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

[41] N. De Cao and W. Aziz. The power spherical distrbution. *Proceedings of the 37th International Conference on Machine Learning, INNF+*, 2020.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[43] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[44] G. Ding, F. Sener, S. Ma, and A. Yao. Every mistake counts in assembly. *arXiv preprint arXiv:2307.16453*, 2023.

[45] N. F. Duarte, M. Raković, J. Tasevski, M. I. Coco, A. Billard, and J. Santos-Victor. Action anticipation: Reading the intentions of humans and robots. *IEEE Robotics and Automation Letters*, 3(4):4132–4139, 2018.

[46] E. Elhamifar and Z. Naing. Unsupervised procedure learning via joint dynamic summarization. In *International Conference on Computer Vision (ICCV)*, 2019.

[47] S. Fahle, C. Prinz, and B. Kuhlenkötter. Systematic review on machine learning (ml) methods for manufacturing processes–identifying artificial intelligence (ai) methods for field application. *Procedia CIRP*, 93:413–418, 2020.

[48] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu. Rmpe: Regional multi-person pose estimation. In *ICCV*, pages 2334–2343, 2017.

[49] J. Feng, R. Xu, J. Hao, H. Sharma, Y. Shen, D. Zhao, and W. Chen. Language models can be logical solvers. *ArXiv*, abs/2311.06158, 2023.

[50] M. Fieraru, M. Zanfir, E. Oneata, A.-I. Popa, V. Olaru, and C. Sminchisescu. Three-dimensional reconstruction of human interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7214–7223, 2020.

[51] A. Flaborea, G. D'Amely, S. D'Arrigo, M. A. Sterpa, A. Sampieri, and F. Galasso. Contracting skeletal kinematics for human-related video anomaly detection, 2023.

[52] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4346–4354, 2015.

[53] J. A. Garcia-Esteban, L. Piardi, P. Leitao, B. Curto, and V. Moreno. An interaction strategy for safe human Co-working with industrial collaborative robots. *Proc. - 2021 4th IEEE Int. Conf. Ind. Cyber-Physical Syst. ICPS 2021*, pages 585–590, 2021.

[54] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *The International Conference on Machine Learning (ICML)*, 2017.

[55] M.-I. Georgescu, A. Bărbălău, R. T. Ionescu, F. Shahbaz Khan, M. Popescu, and M. 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 12737–12747, 2021.

[56] M. I. Georgescu, R. Ionescu, F. S. Khan, M. Popescu, and M. Shah. A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[57] R. Ghoddoosian, I. Dwivedi, N. Agarwal, and B. Dariush. Weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10128–10138, October 2023.

[58] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[59] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. 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*, pages 1705–1714, 2019.

[60] A. Gopalakrishnan, A. Mali, D. Kifer, L. Giles, and A. G. Ororbia. A neural temporal model for human motion prediction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12108–12117, 2019.

[61] J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020.

[62] L. Gualtieri, I. Palomba, E. J. Wehrle, and R. Vidoni. *The Opportunities and Challenges of SME Manufacturing Automation: Safety and Ergonomics in Human–Robot Collaboration*. Springer International Publishing, 2020.

[63] W. Guo, X. Bie, X. Alameda-Pineda, and F. Moreno-Noguer. Multi-person extreme motion prediction with cross-interaction attention. *arXiv preprint arXiv:2105.08825*, 2021.

[64] T. Gupta and A. Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962, June 2023.

[65] T. Gupta and A. Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023.

[66] S. Haddadin, A. Albu-Schaffer, M. Frommberger, J. Rossmann, and G. Hirzinger. The "dlr crash report": Towards a standard crash-testing protocol for robot safety-part i: Results. In *2009 IEEE International Conference on Robotics and Automation*, pages 272–279. IEEE, 2009.

[67] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 733–742, 2016.

[68] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[69] W. Hilal, S. A. Gadsden, and J. Yawney. Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Syst. Appl.*, 193(C), may 2022.

[70] G. Hinton, J. Dean, and O. Vinyals. Distilling the knowledge in a neural network. In *NIPS*, pages 1–9, 2014.

[71] S. Hjorth and D. Chrysostomou. Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly. *Robotics and Computer-Integrated Manufacturing*, 73:102–208, 2022.

[72] J. Ho, A. Jain, and P. 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.

[73] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[74] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[75] Z. Huang, Y. Shen, J. Li, M. Fey, and C. Brecher. A survey on ai-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics. *Sensors*, 21(19):6340, 2021.

[76] N. U. Huda, I. Ahmed, M. Adnan, M. Ali, and F. Naeem. Experts and intelligent systems for smart homes' transformation to sustainable smart cities: A comprehensive review. *Expert Systems with Applications*, 238:122380, 2024.

[77] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *The International Conference on Machine Learning (ICML)*, 2015.

[78] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 448–456, 2015.

[79] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 2014.

[80] ISO. ISO/TS 15066:2016. Robots and robotic devices — Collaborative robots, 2021. https://www.iso.org/obp/ui/#iso:std:iso:ts:15066:ed-1:v1:en.

[81] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317, 2016.

[82] Y. Jain, A. K. Sharma, R. Velmurugan, and B. Banerjee. Posecvae: Anomalous human activity detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2927–2934. IEEE, 2021.

[83] Y. Jang, B. Sullivan, C. Ludwig, I. Gilchrist, D. Damen, and W. Mayol-Cuevas. Epic-tent: An egocentric video dataset for camping tent assembly. In *ICCV*, Oct 2019.

[84] F. Jiang, J. Yuan, S. A. Tsaftaris, and A. K. Katsaggelos. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115(3):323–333, 2011.

[85] R. C. Johnson, K. N. Saboe, M. S. Prewett, M. D. Coovert, and L. R. Elliott. Autonomy and automation reliability in human-robot interaction: A qualitative review. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 53, pages 1398–1402. SAGE Publications Sage CA: Los Angeles, CA, 2009.

[86] A. Kanazawa, J. Kinugawa, and K. Kosuge. Adaptive Motion Planning for a Collaborative Robot Based on Prediction Uncertainty to Enhance Human Safety and Work Efficiency. *IEEE Trans. Robot.*, 35(4):817–832, 2019.

[87] S. Kang, M. Kim, and K. Kim. Safety Monitoring for Human Robot Collaborative Workspaces. *Int. Conf. Control. Autom. Syst.*, 2019-October(Iccas):1192–1194, 2019.

[88] A. M. Kanu-Asiegbu, R. Vasudevan, and X. Du. Bipoco: Bi-directional trajectory prediction with pose constraints for pedestrian anomaly detection. *arXiv preprint arXiv:2207.02281*, 2022.

[89] S. S. Khan and M. G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

[90] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, 2015.

[91] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[92] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations*, ICLR '17, 2017.

[93] M. Knudsen and J. Kaivo-oja. Collaborative robots: Frontiers of current literature. *Journal of Intelligent Systems: Theory and Applications*, 3:13–20, 06 2020.

[94] KUKA. LBR iiwa 14R820 User Manual, 2021. https://www.oir.caltech.edu/twiki_oir/pub/Palomar/ZTF/KUKARoboticArmMaterial/Spez_LBR_iiwa_en.pdf.

[95] G. Lai, H. Liu, and Y. Yang. Learning graph convolution filters from data manifold, 2018.

[96] J. Laplaza, A. Pumarola, F. Moreno-Noguer, and A. Sanfeliu. Attention deep learning based model for predicting the 3d human body pose using the robot human handover phases. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 161–166. IEEE, 2021.

[97] V. LeCun, J. Denker, and S. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, 1989.

[98] K. Lemmerz, P. Glogowski, P. Kleineberg, A. Hypki, and B. Kuhlenkötter. A Hybrid Collaborative Operation for Human-Robot Interaction Supported by Machine Learning. *Int. Conf. Hum. Syst. Interact. HSI*, 2019-June:69–75, 2019.

[99] C. Li, Z. Han, Q. Ye, and J. Jiao. Visual abnormal behavior detection based on trajectory sparse reconstruction analysis. *Neurocomputing*, 119:94–100, 2013.

[100] C. Li, Z. Zhang, W. Sun Lee, and G. Hee Lee. Convolutional sequence to sequence model for human dynamics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[101] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 211–220, 2020.

[102] M. Li, J. Lin, Y. Ding, Z. Liu, J.-Y. Zhu, and S. Han. Gan compression: Efficient architectures for interactive conditional gans. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5283–5293, 2020.

[103] N. Li, F. Chang, and C. Liu. Human-related anomalous event detection via spatial-temporal graph convolutional autoencoder with embedded long short-term memory network. *Neurocomputing*, 490:482–494, 2022.

[104] X. Li and D. Li. Gpfs: A graph-based human pose forecasting system for smart home with online learning. *ACM Trans. Sen. Netw.*, 17(3), 2021.

[105] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. R. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2022.

[106] J. Lim, J. Lee, C. Lee, G. Kim, Y. Cha, J. Sim, and S. Rhim. Designing path of collision avoidance for mobile manipulator in worker safety monitoring system using reinforcement learning. *ISR 2021 - 2021 IEEE Int. Conf. Intell. Saf. Robot.*, pages 94–97, 2021.

[107] S. Liu and W. Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734, 2015.

[108] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6536–6545, 2018.

[109] Y. Liu and S. Chawla. Social media anomaly detection: Challenges and solutions. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2317–2318, 2015.

[110] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2720–2727, 2013.

[111] Z. Lu and E. Elhamifar. Set-supervised action learning in procedural task videos via pairwise order consistency. In *CVPR*, pages 19903–19913, June 2022.

[112] W. Luo, W. Liu, and S. Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 341–349, 2017.

[113] W. Luo, W. Liu, and S. Gao. Normal graph: Spatial temporal graph convolutional networks based prediction network for skeleton based video anomaly detection. *Neurocomputing*, 444:332–337, 2021.

[114] W. Luo, W. Liu, and S. Gao. Normal graph: Spatial temporal graph convolutional networks based prediction network for skeleton based video anomaly detection. *Neurocomputing*, 444:332–337, 2021.

[115] E. Magrini, F. Ferraguti, A. J. Ronga, F. Pini, A. De Luca, and F. Leali. Human-robot coexistence and interaction in open industrial cells. *Robotics and Computer-Integrated Manufacturing*, 61, 2020.

[116] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, 2019.

[117] W. Mao, M. Liu, and M. Salzmann. History repeats itself: Human motion prediction via motion attention. In *The European Conference on Computer Vision (ECCV)*, 2020.

[118] W. Mao, M. Liu, M. Salzmann, and H. Li. Learning trajectory dependencies for human motion prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[119] A. Markovitz, G. Sharir, I. Friedman, L. Zelnik-Manor, and S. 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.

[120] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[121] B. Matthias and T. Reisinger. Example application of ISO/TS 15066 to a collaborative assembly scenario. *47th Int. Symp. Robot. ISR 2016*, 2016:88–92, 2016.

[122] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris. Design considerations for safe human-robot collaborative workplaces. *Procedia CIrP*, 37:248–253, 2015.

[123] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019.

[124] M. Minelli, A. Sozzi, G. De Rossi, F. Ferraguti, F. Setti, R. Muradore, M. Bonfè, and C. Secchi. Integrating model predictive control and dynamic waypoints generation for motion planning in surgical scenario. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3157–3163, 2020.

[125] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng. Large language models as general pattern machines. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, 2023.

[126] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference, 2017.

[127] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. 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.

[128] H. Nascimento, M. Mujica, and M. Benoussaad. Collision avoidance in human-robot interaction using kinect vision system combined with robot's model and data. *IEEE Int. Conf. Intell. Robot. Syst.*, pages 10293–10298, 2020.

[129] D. T. Nguyen, Z. Lou, M. Klar, and T. Brox. Anomaly detection with multiple-hypotheses predictions. In *International Conference on Machine Learning*, pages 4800–4809. PMLR, 2019.

[130] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[131] A. Olmo, S. Sreedharan, and S. Kambhampati. Gpt3-to-plan: Extracting plans from text using gpt-3. *FinPlan 2021*, page 24, 2021.

[132] K. Oono and T. Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

[133] V. Pallagani, B. Muppasani, K. Murugesan, F. Rossi, L. Horesh, B. Srivastava, F. Fabiano, and A. Loreggia. Plansformer: Generating symbolic plans using transformers. *ArXiv*, abs/2212.08681, 2022.

[134] H. Park, J. Noh, and B. Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14372–14381, 2020.

[135] S. H. Park, G. Lee, J. Seo, M. Bhat, M. Kang, J. Francis, A. Jadhav, P. P. Liang, and L.-P. 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.

[136] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[137] J. Pérez, M. Castro, and G. López. Serious games and ai: Challenges and opportunities for computational social science. *IEEE Access*, 2023.

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

[139] M. Petrovich, M. J. Black, and G. Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021.

[140] B. Pistrui, D. Kostyal, and Z. Matyusz. Dynamic acceleration: Service robots in retail. *Cogent Business & Management*, 10(3):2289204, 2023.

[141] B. Prenkaj, D. Aragona, A. Flaborea, F. Galasso, S. Gravina, L. Podo, E. Reda, and P. Velardi. A self-supervised algorithm to detect signs of social isolation in the elderly from daily activity sequences. *Artificial Intelligence in Medicine*, 135:102454, 2023.

[142] Z. Qi, S. Wang, C. Su, L. Su, Q. Huang, and Q. Tian. Self-regulated learning for egocentric video activity anticipation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6715–6730, 2023.

[143] Y. Qian, W. Luo, D. Lian, X. Tang, P. Zhao, and S. Gao. Svip: Sequence verification for procedures in videos. In *CVPR*, pages 19890–19902, June 2022.

[144] F. Ragusa, R. Leonardi, M. Mazzamuto, C. Bonanno, R. Scavo, A. Furnari, and G. M. Farinella. Enigma-51: Towards a fine-grained understanding of human-object interactions in industrial scenarios. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[145] P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol. Ai in health and medicine. *Nature medicine*, 28(1):31–38, 2022.

[146] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

[147] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In M. Meila and T. 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.

[148] J. A. C. Ramon, F. A. C. Herias, and F. Torres. Safe human-robot interaction based on dynamic sphere-swept line bounding volumes. *Robot. Comput. Integr. Manuf.*, 27(1):177–185, 2011.

[149] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks, 2016.

[150] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria. Working together: A review on safe human-robot collaboration in industrial environments. *Ieee Access*, 5:26754–26773, 2017.

[151] R. Rodrigues, N. Bhargava, R. Velmurugan, and S. Chaudhuri. Multi-timescale trajectory prediction for abnormal human activity detection. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[152] D. Rodriguez-Guerra, G. Sorrosal, I. Cabanes, and C. Calleja. Human-Robot Interaction Review: Challenges and Solutions for Modern Industrial Environments. *IEEE Access*, 9:108557–108578, 2021.

[153] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

[154] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4393–4402, 2018.

[155] S. Saadatnejad, A. Rasekh, M. Mofayezi, Y. Medghalchi, S. Rajabzadeh, T. Mordan, and A. Alahi. A generic diffusion-based approach for 3d human pose prediction in the wild. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[156] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017.

[157] A. Sampieri, G. M. D. di Melendugno, A. Avogaro, F. Cunico, F. Setti, G. Skenderi, M. Cristani, and F. Galasso. Pose forecasting in industrial human-robot collaboration. In *Computer Vision – ECCV 2022*, pages 51–69, Cham, 2022. Springer Nature Switzerland.

[158] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[159] F. Sener, D. Chatterjee, D. Shelepov, K. He, D. Singhania, R. Wang, and A. Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. *CVPR*, 2022.

[160] A. Shah. Media and artificial intelligence: Current perceptions and future outlook. *Academy of Marketing Studies Journal*, 28(2), 2024.

[161] A. Shah, B. Lundell, H. Sawhney, and R. Chellappa. Steps: Self-supervised key step extraction and localization from unlabeled procedural videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10375–10387, October 2023.

[162] J. Shah, J. Wiken, C. Breazeal, and B. Williams. Improved human-robot team performance using Chaski, a human-inspired plan execution system. *HRI 2011 - Proc. 6th ACM/IEEE Int. Conf. Human-Robot Interact.*, pages 29–36, 2011.

[163] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua. Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[164] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso. Space-time-separable graph convolutional network for pose forecasting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11209–11218, 2021.

[165] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[166] Y.-F. Song, Z. Zhang, C. Shan, and L. Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022.

[167] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6479–6488, 2018.

[168] D. Surís, S. Menon, and C. Vondrick. Vipergpt: Visual inference via python execution for reasoning. *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023.

[169] B. Tang, M. A. Lin, I. Akinola, A. Handa, G. S. Sukhatme, F. Ramos, D. Fox, and Y. Narang. Industreal: Transferring contact-rich assembly tasks from simulation to reality. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[170] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1207–1216, 2019.

[171] D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54:45–66, 2004.

[172] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano. Human motion diffusion model. In *The 11th International Conference on Learning Representations*, 2023.

[173] H. Thanh-Tung and T. Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2020.

[174] C. Torkar, S. Yahyanejad, H. Pichler, M. Hofbaur, and B. Rinner. Rnn-based human pose prediction for human-robot interaction. In *Proceedings of the ARW & OAGM Workshop 2019*, pages 76–80, 2019.

[175] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[176] H. Tu, C. Wang, and W. Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *European Conference on Computer Vision*, pages 197–212. Springer, 2020.

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

[178] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. 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.

[179] L. Vianello, J.-B. Mouret, E. Dalin, A. Aubry, and S. Ivaldi. Human posture prediction during physical human-robot interaction. *IEEE Robotics and Automation Letters*, 2021.

[180] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

[181] C. Wang, Y. Wang, Z. Huang, and Z. Chen. Simple baseline for single human motion forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2260–2265, 2021.

[182] X. Wang, T. Kwon, M. Rad, B. Pan, I. Chakraborty, S. Andrist, D. Bohus, A. Feniello, B. Tekin, F. V. Frujeri, N. Joshi, and M. Pollefeys. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *ICCV*, pages 20270–20281, October 2023.

[183] X. Wang, S. Zhang, Z. Qing, Y. Shao, Z. Zuo, C. Gao, and N. Sang. Oadtr: Online action detection with transformers. In *ICCV*, pages 7565–7575, 2021.

[184] Z. Wang, Z. Chen, J. Ni, H. Liu, H. Chen, and J. Tang. Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3726–3734, 2021.

[185] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. hsin Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022.

[186] J. W. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, and T. Ma. Larger language models do in-context learning differently. *ArXiv*, abs/2303.03846, 2023.

[187] Y.-H. Weng, C.-H. Chen, and C.-T. Sun. Toward the human–robot co-existence society: On safety intelligence for next generation robots. *International Journal of Social Robotics*, 1:267–282, 2009.

[188] J. Wyatt, A. Leach, S. M. Schmon, and C. 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.

[189] Z. Xiao, K. Kreis, and A. Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022.

[190] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[191] C. Xin, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, J. Yu, and G. Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[192] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018.

[193] S. Xu, Y.-X. Wang, and L.-Y. Gui. Diverse human motion prediction guided by multi-level spatial-temporal anchors. In *European Conference on Computer Vision (ECCV)*, pages 251–269, 2022.

[194] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32, Apr. 2018.

[195] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, M. Yang, and B. Cui. Diffusion models: A comprehensive survey of methods and applications. *CoRR*, abs/2209.00796, 2022.

[196] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *The European Conference on Computer Vision (ECCV)*, 2020.

[197] Y. Yuan and K. Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[198] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

[199] J. Zhang, H. Liu, Q. Chang, L. Wang, and R. X. Gao. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. *CIRP annals*, 69(1):9–12, 2020.

[200] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and F. Taher. Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access*, 2022.

[201] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1933–1941, New York, NY, USA, 2017. Association for Computing Machinery.

[202] Y. Zhao and Y. Dou. Pose-forecasting aided human video prediction with graph convolutional networks. *IEEE Access*, 8:147256–147264, 2020.

[203] C. Zhong, L. Hu, Z. Zhang, Y. Ye, and S. Xia. Spatio-temporal gating-adjacency gcn for human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6447–6456, June 2022.

[204] Y. Zhong, L. Yu, Y. Bai, S. Li, X. Yan, and Y. Li. Learning procedure-aware video representation from instructional videos and their narrations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, jun 2023.

[205] Y. Zhong, L. Yu, Y. Bai, S. Li, X. Yan, and Y. Li. Learning procedure-aware video representation from instructional videos and their narrations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14825–14835, Los Alamitos, CA, USA, jun 2023. IEEE Computer Society.

[206] L. Zhou, Y. Du, and J. Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5826–5835, October 2021.

[207] D. Zhukov, J.-B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019.