Understanding Videos by Learning Structured, Robust and Efficient Representations

Ayush Kumar Rai, MSc

Co-supervised by Prof. Alan F. Smeaton and

Prof. Noel E. O'Connor

A Dissertation submitted in fulfilment of the requirements for the award of Doctor of Philosophy (Ph.D.)

School of Computing

Dublin City University

August 2025

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Ayush Kumar Rai ID No.: 20211625 Date: 12/08/2025

Acknowledgements

A research contribution is not the achievement of the student author alone but rather a collective effort of collaborators and team working behind the scene. This PhD research would not have been possible without the support of many people, to whom I would like to express my gratitude.

First and foremost, I would like to thank my supervisors, Prof. Alan F. Smeaton and Prof. Noel E. O'Connor, for their invaluable guidance, encouragement, and expertise throughout this research journey. To both of you, thank you for providing me an opportunity to pursue doctoral studies at Insight DCU during the uncertain times of COVID-19. The constructive feedback, discussion and writing tips during our meeting sessions helped me to polish and refine my research ideas, which was instrumental in molding me into a more competent researcher.

I am especially grateful to my Independent Panel Member, Kevin McGuinness, for his technical insights, rigorous discussions and providing me with intuitive explanations on complex topics. Also thank you for presenting my first PhD paper in Hawaii. You have had a very profound influence on me as a researcher on approaching problems from fundamental principles, and I will always cherish it.

I would also like to extend my heartfelt thanks to my closest collaborators, Tarun Krishna and Feiyan Hu, for their dedication and shared efforts who made this journey both productive and enjoyable.

I am thankful for the industrial collaborations that enriched my research and opened multiple avenues for me to explore. Special thanks to Alexandru Drimbarean from Tobii (Xperi), and Kyle Min from Intel Labs, for their guidance, contributions

and support.

I would also like to acknowledge Paul Albert, Luis Lebron, and Julia Dietlmeier for their assistance, as well as Deirdre McCabe and Deirdre Sheridan for their support on the administrative side of things.

Finally, my deepest gratitude goes to my parents and family, whose unwavering encouragement and patience has been the foundation of my success.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

ML ..... Machine Learning

DL ..... Deep Learning

CV ..... Computer Vision

NLP ..... Natural Language Processing

NN ..... Neural Network

MLP ..... Multi Layer Perceptron

DNN ..... Deep Neural Network

FNN ..... Feed-Forward Network

MSE ..... Mean Squared Error

MHSA ..... Multi-Head self-attention

SGD ..... Stochastic Gradient Descent

ReLU ..... Rectified Linear Unit

GELU ..... Gaussian Error Linear Unit

CNN ..... Convolutional Neural Network

RNN ..... Recurrent Neural Network

LSTM ..... Long Short Term Memory

AE ..... Autoencoder

GRU ..... Gated Recurrent Unit

I3D ..... Inflated 3D Convolutional Networks

TSN ..... Temporal Segment Network

TRN ..... Temporal Relation Network

TSM ..... Temporal Shift Module

NCE ..... Noise Constrastive Estimation

VTN ..... Video Transformer Network

ViT ..... Vision Transformer

LN ..... Layer Normalization

SSL ..... Self-supervised learning

KL ..... Kullback-Leibler Divergence

VAE ..... Variational Autoencoders

ID ..... Instance Discrimonation

OCC ..... One Class Classification

SOTA ..... State-of-the-art

OOD ..... Out of Distribution

LDMs ..... Latent Diffusion Models

PSNR ..... Peak Signal to Noise Ratio

MHSA ..... Multi-head Self Attention

GEBD ..... Generic Event Boundary Detection

VAD ..... Video Anomaly Detection

OCC ..... One Class Classification

PA/PAs ..... Pseudo-Anomaly/Pseudo-Anomalies

MAE ..... Masked Autoencoder

MIM ..... Masked Image Modeling

MVM ..... Masked Video Modeling

PPO ..... Proximal Policy Optimisation

Publications

Publications arising directly from this thesis

- ***Ayush K. Rai**, *Kyle Min, Tarun Krishna, Feiyan Hu, Alan F. Smeaton, Noel E. O'Connor. Reinforcement Learning meets Masked Video Modeling : Trajectory-Guided Adaptive Token Selection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, 2025.

- **Ayush K. Rai**, Tarun Krishna, Feiyan Hu, Alexandru Drimbarean, Kevin Mcguinness, Alan F. Smeaton, Noel E. O'Connor. Video Anomaly Detection via Spatio-Temporal Pseudo-Anomaly Generation : A Unified Approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2024, pp. 3887-3899.

- **Ayush K. Rai**, Tarun Krishna, Julia Dietlmeier, Kevin McGuinness, Alan F. Smeaton, Noel E. O'Connor. Motion Aware Self-Supervision for Generic Event Boundary Detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2023, pp. 2728-2739.

Other publications by the author

- Tarun Krishna, **Ayush K. Rai**, Alexandru Drimbarean, Eric Arazo, Paul Albert, Alan F. Smeaton, Kevin McGuinness, Noel E. O'Connor. Unifying Synergies between Self-supervised Learning and Dynamic Computation. In Proceedings of British Machine Vision Conference (BMVC), 2023.

- *Tarun Krishna, ***Ayush K. Rai**, Yasser A. D. Djilali, Alan F. Smeaton, Kevin McGuinness, Noel E. O'Connor. Dynamic Channel Selection in Self-Supervised Learning. In Proceedings of Irish Machine Vision and Image Processing Conference (IMVIP) August 2022, pp 121-128.

(* Equal contribution).

Understanding Videos by Learning Structured, Robust and Efficient Representations

Ayush K. Rai

Abstract

With an enormous volume of unstructured video content constantly being generated online, designing intelligent systems for automatic understanding of visual data could have a direct and beneficial effect on several fields such as real-world surveillance, robotics, healthcare, entertainment, content retrieval etc. However, extracting meaningful and relevant information from videos still remains a challenging task and an open area of research. Learning powerful representations in the video domain involves multiple facets such as structural feature learning, modeling motion, multi-modal feature learning, feature disentanglement etc., with the primary goal of holistic video understanding. Recently, self-supervised learning has gained prominence as an effective paradigm for representation learning in images and videos, eliminating the need for additional label annotations. The objective of this thesis is to thoroughly investigate various video modeling techniques, primarily aimed at learning structured, robust, and efficient video representations within the framework of self-supervised learning.

To focus on learning *structured* video representations, this work first addresses the task of generic event boundary detection by revisiting a self-supervised method and enhancing it by incorporating a differentiable motion estimation module to capture the generic spatial and temporal diversities in the videos. Extensive experiments on the Kinetics-GEBD and TAPOS datasets demonstrate the efficacy of the proposed approach compared to the other self-supervised state-of-the-art methods.

In order to embed *robustness* into learned video representations, the thesis then tackles the problem of video anomaly detection from the perspective of recognizing out of distribution samples. A novel method is proposed to generate spatio-temporal pseudo-anomalies by inpainting masked image regions with a pre-trained Latent Diffusion Model and perturbing optical flow using mixup to simulate spatio-temporal distortions. Additionally, a unified framework is introduced to detect real-world anomalies under the one-class classification setting by learning three anomaly indicators: reconstruction quality, temporal irregularity, and semantic inconsistency. Rigorous evaluations on Ped2, Avenue, ShanghaiTech, and UBnormal benchmarks highlight the method's effectiveness compared to existing state-of-the-art approaches.

To learn video representations *efficiently*, this research proposes a novel and generalizable Trajectory-Aware Adaptive Token Sampler (TATS) module that learns to adaptively sample motion-centric tokens for masked autoencoder (MAE) pre-training by modeling their motion trajectories in videos. Additionally, a unified training recipe is also introduced that facilitates the joint optimization of both MAE and TATS from scratch using Proximal Policy Optimization to ensure stable convergence during pre-training even with aggressive masking. Comprehensive evaluation on benchmark datasets (Kinetics-400, Something-Something v2, UCF101, HMDB51) for action recognition demonstrates the effectiveness, generalization, transferability, and efficiency of our work compared to the state-of-the-art methods.

Chapter 1

Introduction

1.1 Motivation

Recently, Computer Vision (CV) has witnessed a great deal of progress due to the development of advanced Deep Learning (DL) based models, which are very effective at extracting and learning meaningful representations from images or videos. DL has achieved remarkable breakthroughs in traditionally challenging tasks such as image classification ([He et al., 2016], [Krizhevsky et al., 2012a]), and object detection ([Girshick, 2015], [Ren et al., 2015a]). These techniques have also led to a great breakthrough in video understanding tasks such as action anticipation [Miech et al., 2019a, Abu Farha et al., 2018], temporal action detection [Chao et al., 2018, Gao et al., 2017], temporal action segmentation [Lea et al., 2016a, Kuehne et al., 2014] and temporal action parsing [Pirsiavash and Ramanan, 2014, Shao et al., 2020a].

The remarkable success of DL across various domains is largely dependent on the availability of large-scale annotated datasets. However, acquiring annotations is costly and labor-intensive, which poses an even greater challenge for video data. Moreover, the use of human-generated annotations often leads to models with biased learning and poor domain generalization and robustness. As an alternative, self-supervised learning (SSL) [Oord et al., 2018, Chen et al., 2020b, Dave et al., 2022] provides a way for representation learning which does not require annotations and has shown promise in both image and video domains. Unlike the image

domain, learning video representations is more challenging due to the presence of the extra temporal dimension and motion dynamics in the videos. However, these challenges also provide opportunities for exploring novel research ideas to enhance representation in the video domain.

The fundamental step in designing a DL pipeline for video understanding tasks involves a vital step of video feature extraction referred to as *video representation learning*. To acquire more expressive, generalizable, and transferable representations, harnessing the potential of SSL for end-to-end video representation learning has shown significant potential. SSL enables pre-training a single model to learn foundational features, which can then be finetuned for various downstream tasks. SSL also allows for designing pretext tasks to incorporate specific desired properties into the model while eliminating the need to learn separate models for different tasks. There are various video-specific SSL pretext tasks that can be used including temporal ordering [Fernando et al., 2017, Lee et al., 2017, Misra et al., 2016, Wei et al., 2018, Wang et al., 2019a], future prediction [Vondrick et al., 2016, Mathieu et al., 2016, Lotter et al., 2017, Vondrick et al., 2018, Diba et al., 2019], spatiotemporal contrast [Feichtenhofer et al., 2021, Han et al., 2019, Qian et al., 2021b, Sun et al., 2019], temporal coherence [Goroshin et al., 2015, Wiskott and Sejnowski, 2002] object motion [Agrawal et al., 2015, Pathak et al., 2017, Wang and Gupta, 2015, Wang et al., 2019c], and masked modeling [Tong et al., 2022, He et al., 2022a].

Though many challenging topics in video representation learning remain underexplored, two are particularly important: *(1) What makes a good video representation and (2) What properties should a video representation have?* A desirable video representation should have a number of characteristics. It should be:

- Structured video representation implies modeling spatial diversities, fine-grained temporal coherency, long range temporal dependencies and learning motion patterns in the video.
- Robustness in video representation indicates the ability to remain resistant to spatio-temporal variations such as changes in lighting, background noise,

and visual clutter, while maintaining strong focus on the important aspects elements such as actions, motion patterns, and human–object interactions.

- Efficient (adaptive) video representation refers to the idea of learning only the relevant information while discarding redundant content, using computationally less expensive strategies by augmenting deep learning models for videos with adaptive computation techniques.

- Disentangled video representation involves decomposing video representation into semantically meaningful factors such as objects, entities, inter-object relationships, contextual information etc.

- Causal video representations capture the cause and effect relationships within the video and not just correlations or visual patterns. The goal is to understand why certain events happen, rather than just recognizing what is happening.

These characteristics are essential for learning generic features that can generalize effectively to unseen datasets. In this thesis, we focus on learning *structured, robust, and efficient representations* in a self-supervised setting, as highlighted above.

Structured Video Representation Learning. Many widely studied tasks in video understanding, such as temporal action detection [Chao et al., 2018, Gao et al., 2017], temporal action segmentation [Lea et al., 2016a, Kuehne et al., 2014] and temporal action parsing [Pirsiavash and Ramanan, 2014, Shao et al., 2020a], are predominantly addressed using temporally local (processing short intervals of time) techniques, which do not sufficiently capture the structure of the video. This raises a fundamental question: *Is there a canonical approach to summarizing a video representation?*. To put this in another way: *Can we inherently learn a video's structure by capturing temporally granular (fine-grained) and temporally persistent (global) features while leveraging motion patterns?* The goal of structured video representation learning is to develop novel DL models to address these issues. In this thesis, generic event boundary detection (GEBD) [Shou et al., 2021, Kang et al.,

2021b] is chosen as an appropriate downstream task for understanding this aspect of video representations (this is further explained in Chapter 2 and Chapter 3).

Robust Video Representation Learning. Video representations must be resilient to spatio-temporal perturbations, such as lighting variations, background noise, and clutter, while ensuring strong attentiveness to the relevant information (action, motion patterns, human-object interaction) within the videos. To explore robust video representation we investigate the problem of video anomaly detection (VAD) [Ramachandra et al., 2020b] as detailed in Chapter 2 and Chapter 4. Learning a robust representation ensures that the model filters out irrelevant noise, such as lighting variations and background clutter while remaining highly sensitive to meaningful deviations caused by real-world anomalies (unusual object or activity in the scene). VAD is an ideal downstream task for investigating solutions to this whilst effectively generalising across different datasets and unseen real-world anomalies.

Efficient Video Representation Learning. Learning video representations efficiently remains a significant challenge for DL models due to their heavy reliance on large computational resources. While capturing spatio-temporal features in videos can enhance generalization across various downstream video understanding tasks, much of this information is highly redundant and needs to be filtered out while preserving the relevant details, resulting in computationally efficient pre-training of the DL model. To accomplish this, integrating adaptive computation [Veit and Belongie, 2018, Li et al., 2021a, Meng et al., 2020] with the SSL pre-training objective proves to be highly effective. In order to learn efficient video representation, we incorporate a learnable and adaptive token sampler module into the masked video modeling (MVM) [Tong et al., 2022] objective as described in Chapter 5.

1.2 Hypotheses and Research Questions

As explained above, in this thesis we focus on learning *structured, robust, and efficient representations* in a self-supervised setting. Our four hypotheses revolve around learning these characteristics of good video representations.

Hypothesis 1 ($\mathbf{H_1}$): We hypothesize that a *structure* can be embedded into a video representation by designing relevant self-supervised pretext tasks that can model spatial diversities, fine-grained temporal coherency, long range temporal dependencies and motion patterns in videos. The downstream task selected to evaluate structured video representations is Generic Event Boundary Detection, GEBD (in Chapter 2, 3), which aims to detect moments in videos that are naturally perceived by humans as generic and taxonomy-free event boundaries. GEBD is well suited to validate whether a structure has been encoded into a learned video representation or not due to the generic nature of event boundaries. This leads us to research questions R_1 and R_2.

$\mathbf{R_1}$: How can we leverage the power of SSL to capture spatio-temporal diversities and relationships involved in videos?

$\mathbf{R_2}$: How can we develop an SSL framework for video understanding that accounts for both appearance and motion features? Do we need an explicit motion-specific training objective, or can this be implicitly achieved?

Hypothesis 2 ($\mathbf{H_2}$): We hypothesize that *robustness* to spatio-temporal perturbations can be instilled into the learned video representation by exposing the DL models to near-distribution samples (samples lying on the periphery of the data distribution) during the self-supervised pre-training step while maintaining their sensitivity to real-world anomalies such as unusual object or activity in the scene. We aim to examine the robustness attribute of video representation by exploring the problem of video anomaly detection, VAD (Chapter 2,4) under the one-class classification (OCC) setting. Here, the OCC setting refers to a scenario where the training data consists of videos containing only normal instances, while the test data includes both normal and anomalous instances. Also in the context of VAD, we refer near-distribution samples as pseudo-anomalies (PAs). This provides the basis for research questions R_3 and R_4.

R_3 : Is it possible to synthetically generate generic PAs by introducing spatio-temporal distortions into normal data in order to detect real-world anomalies effectively? Furthermore, can such PAs transfer across multiple VAD datasets?

R_4 : How can we design a VAD pipeline that aggregates different anomaly indicators to create a unified anomaly scoring mechanism that effectively captures spatial, temporal, and semantic inconsistencies?

It should be noted in this context that a robust representation does not mean that the model ignores anomalies. Rather, learning a robust representation enables a model to filter out irrelevant noise while maintaining high sensitivity to meaningful deviations caused by real-world anomalies. Furthermore, it should generalize well across different datasets and unseen anomalies.

Hypothesis 3 (H_3) : We hypothesize that integrating adaptive computation strategies into the self-supervised training objective can facilitate the learning of more transferable and generalizable video representations in a more *efficient* manner compared to those learned with static computation. More specifically, adaptive computation in this context refers to dynamic token sampling of the most informative space-time tokens in videos, while the self-supervised objective is based on MVM. To assess the quality of learned video representation, we choose the downstream task of action recognition on benchmark datasets. This leads us to research questions R_5 and R_6.

R_5 : How can we incorporate adaptive computation in a self-supervised pre-training objective such as MVM to dynamically select informative space-time tokens based on the given input?

R_6 : Are representations learnt through dynamic computation (adaptive masking) as transferable to downstream tasks (action recognition) as the ones learnt with static computation (random masking)?

1.3 Structure of the Thesis

This thesis is structured as follows. Chapter 2 presents the technical background necessary for understanding the research presented in the thesis and also provides a high-level overview of related work.

Chapter 3 delves into the concept of learning *structured* video representations within the self-supervised learning (SSL) framework. Specifically, we examine $\mathbf{H_1}$ (R_1, R_2), which suggests that designing effective self-supervised pretext tasks can embed spatial diversity, fine-grained temporal coherence, long-range temporal dependencies, and motion patterns into the learned model. To validate $\mathbf{H_1}$, we propose a self-supervised approach that integrates frame-level and clip-level pretext tasks, along with a differentiable motion learning module, and assess its performance on the GEBD task (Chapter 2). This chapter concludes by demonstrating that the structured representation learned by our self-supervised framework achieves comparable performance to state-of-the-art methods on this challenging task. The proposed motion-aware self-supervised approach achieves comparable performance to other self-supervised state-of-the-art methods for generic event boundary detection while being significantly simpler than prior methods in terms of architectural complexity and it learns general motion features without explicit motion pretext tasks.

Chapter 4 explores how to incorporate *robustness* into learned video representations within a self-supervised setting. Specifically, this chapter investigate $\mathbf{H_2}$ (R_3, R_4), which presumes that robustness to spatio-temporal perturbations can be achieved by exposing the model to near-distribution samples during SSL pre-training while maintaining sensitivity to real-world anomalies. To validate $\mathbf{H_2}$, we examine the VAD task within the OCC setting. Using available normal data, we generate PAs through generative models such as diffusion models [Rombach et al., 2022] or by applying mixup augmentation [Zhang et al., 2018] to distort optical flow [Zach et al., 2007], incorporating them into the VAD framework's pre-training. The spatial, temporal, and semantic information extracted from PAs further enables the aggregation of multiple anomaly indicators, enhancing real-world video anomaly de-

tection. Our model achieves competitive anomaly detection performance compared to state-of-the-art reconstruction-based methods. The proposed approach achieves comparable performance to state-of-the-art methods while avoiding strong inductive biases and demonstrating transferability across multiple datasets.

Chapter 5 investigates *efficient* video representation learning. Specifically, we investigate $\mathbf{H_3}$ (R_5, R_6), which hypothesizes that integrating adaptive computation strategies into the self-supervised training objective can facilitate the learning of transferable and generalizable video representations more efficiently than static computation methods. To assess $\mathbf{H_3}$, we propose a Trajectory-Aware Adaptive Token Sampler module that dynamically selects the most relevant motion-centric space-time tokens for the self-supervised pre-training objective of MVM. The effectiveness of the learned representation is evaluated through the downstream task of action recognition on benchmark datasets. The proposed approach delivers effective and generalizable action recognition across multiple benchmark datasets, outperforming state-of-the-art masking methods while remaining memory-efficient.

Finally, Chapter 6 provides a summary of the research conducted in this thesis. The results and findings are discussed by relating them to the hypotheses and research questions presented in this thesis. We conclude with some suggestions for possible future work and some general remarks.

Chapter 2

Background

This Chapter provides the theoretical and technical background necessary for understanding the research discussed in the thesis. In particular, this Chapter introduces fundamental terminologies and concepts in deep learning for computer vision, while also explaining the technical details needed for understanding video representation learning. Section 2.1 provides an overview of key concepts for representation learning in machine learning and deep learning. Section 2.2 and Section 2.3 discuss various strategies for extracting video representations under the supervised and self-supervised framework respectively. Section 2.5 outlines different downstream tasks used in the thesis while Section 2.6 provides details about the datasets used for experimentation and evaluation.

2.1 Background in Machine Learning

In this Section, we review fundamental concepts in ML and DL essential for understanding the research discussed in this thesis.

2.1.1 Supervised Learning

Supervised machine learning involves training a system to automatically predict an output given an input, based on a set of labeled examples. If the output value is a continuous quantity then it is a regression problem, while if the output value is

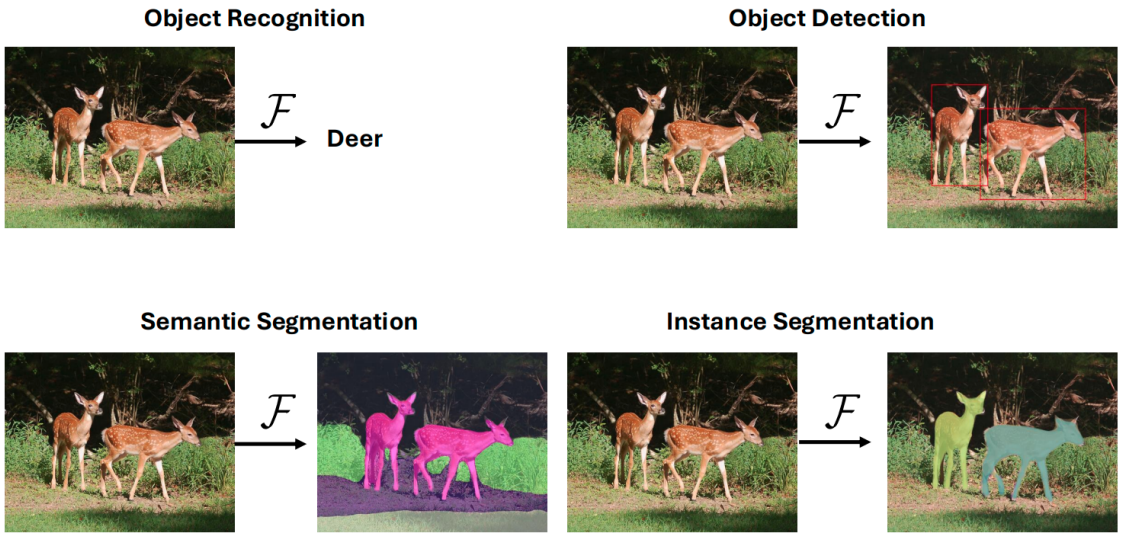


Figure 2.1: **Examples of Supervised Learning Problems in Computer Vision** - (Top-Left) Object Recognition: The task is to predict which object is present in the input image. (Top Right) Object Detection: The task is to detect objects by predicting the bounding box locations as well as the object categories. (Bottom Left) Semantic Segmentation: The task involves predicting the pixel mask for each detected object. (Bottom Right) Instance Segmentation: Extends semantic segmentation by distinguishing between individual object instances within the same object class.

discrete class label, then it is a classification problem. In this section, we focus on the classification task however the explanations are easily transferable to the regression task. Some of the supervised learning problems in computer vision are illustrated in Figure 2.1.

Problem setup. $\mathbf{x} \subset \mathcal{X}$ and $\mathbf{y} \subset \mathcal{Y}$ are assumed to be the input and the output of the supervised learning model, where $\mathcal{X}$ and $\mathcal{Y}$ represent the input and output space respectively. For example, in the case of object recognition, $\mathbf{x}$ is an image and $\mathbf{y}$ is the object category index whose values are ranging from 1 to C where C is the number of classes.

The training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{Y}$ is composed of N training samples. The goal is to learn a mapping $f \subset \mathcal{F}$ that can correctly predict the label $\mathbf{y}$ given the input $\mathbf{x}$, where $\mathcal{F}$ represents the set of all functions. The predicted label $\hat{\mathbf{y}}$ is the output of the machine learning model f and is obtained using the following equations:

$$f(\mathbf{x}) = \hat{\mathbf{y}} \tag{2.1}$$

$$\hat{\mathbf{y}} = \underset{c}{\operatorname{argmax}}\, \hat{\mathbf{y}}^c \tag{2.2}$$

where $\hat{\mathbf{y}}$ is a C-dimensional prediction vector which can be used to obtain a probability distribution over the set of possible classes such that $\Sigma_{c=1}^{C}\hat{\mathbf{y}}^c = 1$ and $\mathbf{y}^c > 0$ $\forall c = 1...C$. The value $\hat{\mathbf{y}}^c$ corresponds to the probability that the input $\mathbf{x}$ belongs to the c-th category. The ML model or the mapping function f is learned using the training examples from $\mathcal{D}$ with the underlying goal of generalizing to unseen examples.

Neural Network. We restrict the mapping function or the machine learning model f to *feedforward neural networks* since all the approaches proposed in this thesis belong to this category. Neural networks are composed of many different functions or layers, where each layer is itself a neural network. The term feedfoward means that the information flows strictly in a forward direction from the input to the output, such as shown in Figure 2.2. The mapping function f is composed of n layers and is parameterized such that:

$$f(\mathbf{x}) = f_\theta(\mathbf{x}) = f_{\theta_n}(f_{\theta_{n-1}}(...f_{\theta_k}(...f_{\theta_1}(\mathbf{x})...)...)) \tag{2.3}$$

$$\boldsymbol{h_k} = f_{\theta_k}(\boldsymbol{h_{k-1}}) \tag{2.4}$$

where $\theta = \{\theta_1, \theta_2, ..., \theta_n\} \in \Theta$ is the set of trainable parameters of f, Θ is the parameters space and f_k is the k-th layer whose input, output and trainable parameters are given by h_{k-1}, h_k, θ_k respectively. Since f is parametrized by the trainable parameters θ, we use f_θ for denoting the mapping function. In other words, a feedfoward neural network f_θ can also be seen as acyclic directed graph as shown in Figure 2.2.

Optimization. In order to learn the function f_θ a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$

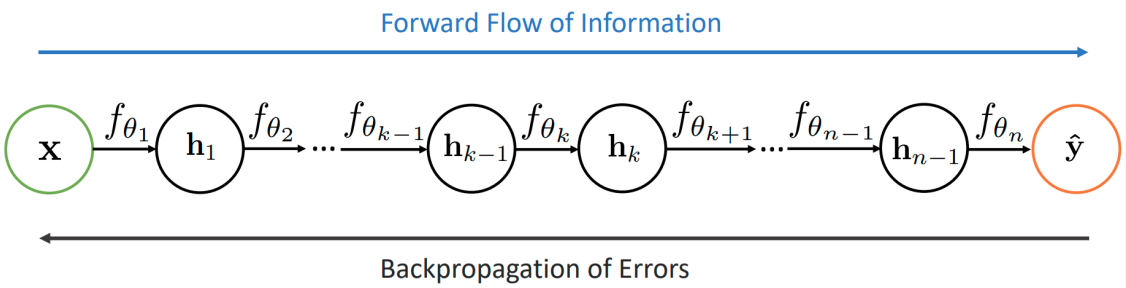


Figure 2.2: **Feedforward Neural Network.** A mapping function f is a feedforward neural network that can be composed of n layers or functions. The green node corresponds to the input while the orange node is the output. The information flows feedforward (from left to right) for producing an output given an input. And the system is trained by backpropagating the error in a backwards manner (from right to left).

is defined on data points corresponding to the cost of predicting $\hat{\mathbf{y}}$ when the label is actually $\mathbf{y}$. Most commonly used loss function in the classification task is the cross-entropy loss [Le Cun et al., 1997] given by:

$$\mathcal{L}_{CE}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{c=1}^{C} \mathbf{y}^c log \hat{\mathbf{y}} \tag{2.5}$$

where $\mathbf{y}$ is a one-hot vector representation of the groundtruth class such that $\mathbf{y} \in \{0,1\}^C$ and $\sum_{c=1}^{C} \mathbf{y}^c = 1$.

In order to learn the most optimal f_θ, we follow the principle of Empirical Risk Minimization (ERM) where the risk is defined as the expectation of the loss function $\mathcal{L}$. Since we do not know the joint distribution of the data points we cannot compute the true risk. Instead, we minimize the empirical risk by averaging the output of the loss function $\mathcal{L}$ on the training set $\mathcal{D}$. Hence, the optimization problem for obtaining the set of most optimal parameters θ^* is of the general form:

$$\mathcal{J}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y}) + \mathcal{R}(\theta) \tag{2.6}$$

$$\theta^* = \underset{\theta\in\Theta}{\operatorname{argmin}}\, \mathcal{J}(\theta) \tag{2.7}$$

where $\mathcal{J}$ is the objective function composed of the loss function $\mathcal{L}$ and $\mathcal{R}$ is a regularization term. This optimization problem is a non-convex optimization problem due

to the presence of non-linear functions in f_θ. In practice, Stochastic Gradient Descent (SGD) and the backpropagation rule is availed for estimating the parameters as described below.

Stochastic Gradient Descent. The standard approach to minimizing an objective function $\mathcal{J}$ over a training set $\mathcal{D}$ in the context of neural networks is to use stochastic gradient descent (SGD). This requires that the model f_θ be differentiable with respect to all parameters in θ. Basically the method involves computing the gradient of the objective function $\frac{\partial \mathcal{J}(\theta)}{\partial \theta}$ on the training set $\mathcal{D}$ and updating the parameters in the opposite direction of the gradient.

However, computing the exact gradient of the objective function can be computationally expensive when the training set is large. To address this, a common approach is to use a variant known as mini-batch stochastic gradient descent (SGD). This method replaces the true gradient with an estimate $\frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta}$ computed over a mini-batch $\mathcal{S}$, which is a subset of examples randomly sampled from the full training set $\mathcal{D}$ such that:

$$\frac{\partial \mathcal{J}(\theta)}{\partial \theta} \approx \frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\left[\frac{1}{|\mathcal{S}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}} \mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y}) + \mathcal{R}(\theta)\right] \tag{2.8}$$

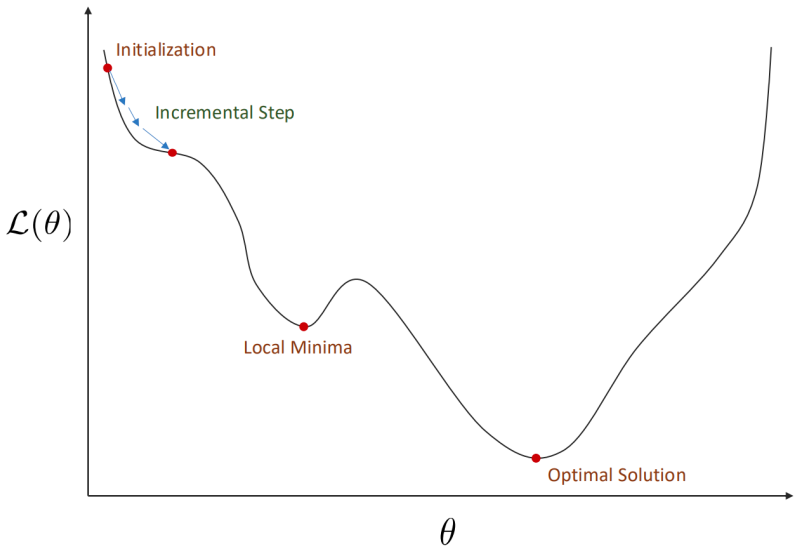


Figure 2.3: Illustration of the optimization process of the gradient descent method. The optimization can get stuck in a local minimum as it is dependent on the initialization.

To ensure that $\frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta}$ is a good approximation of the true gradient $\frac{\partial \mathcal{J}(\theta)}{\partial \theta}$, the mini-batch $\mathcal{S}$ must be representative of the full training set $\mathcal{D}$. This can be achieved by selecting a sufficiently large number of training examples in $\mathcal{S}$. The parameters are then updated in the direction opposite to the estimated gradient according to the following equation:

$$\theta \leftarrow \theta - \eta \frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta} \tag{2.9}$$

where η is the learning rate.

Mini-batch SGD is an iterative process that begins with random initialization of the trainable parameters in θ. An epoch consists of a complete pass over all mini-batches. The training process is repeated across multiple epochs until convergence is achieved on the training set. Both the learning rate η and the initialization values of the parameters are crucial for the training procedure. If the learning rate is too small, training can be slow and may get stuck in local minima due to poor initialization. On the other hand, if the learning rate is too large, the optimization may fail to converge. Figure 2.3 illustrates an example of the iterative nature of the gradient descent algorithm. To ensure generalization on the unseen data, a validation set is often used to monitor progress. The training process stops when the performance metric computed on the validation set no longer improves or reaches a plateau.

Backpropagation. Training the machine learning model f_θ using mini-batch SGD involves calculating the gradient $\frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta}$, as shown in Equation 2.9. However, computing $\frac{\partial \mathcal{L}(f_\theta(\mathbf{x}),\mathbf{y})}{\partial \theta}$, which is a part of the overall gradient, can become computationally expensive as f_θ grows deeper. One approach to address this challenge is to apply the chain rule and compute the gradient layer by layer, as follows:

$$\frac{\partial \mathcal{L}(f_\theta(\mathbf{x}),\mathbf{y})}{\partial \theta_l} = \frac{\partial \mathcal{L}(f_\theta(\mathbf{x}),\mathbf{y})}{\partial h_n} \left(\prod_{k=l+1}^{n} \frac{\partial \boldsymbol{h_k}}{\partial h_{k-1}} \right) \frac{\partial \boldsymbol{h}_l}{\partial \theta_l} \tag{2.10}$$

This principle is known as the backpropagation rule and consists of an iterative backward propagation of errors from the last layer such as shown in Figure 2.2.

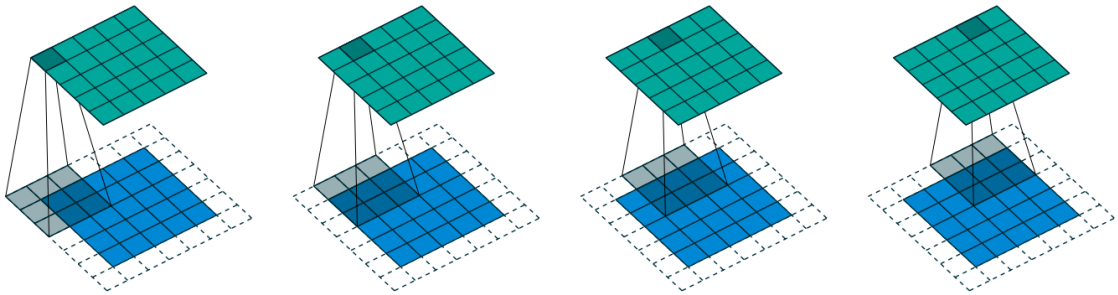


Figure 2.4: Depiction of a convolution operation on grid-like structure (could also be thought of as image pixels). Figure from [Dumoulin and Visin, 2016]

Convolutional Neural Networks (CNN). Convolutional layers [LeCun, 1998] are a fundamental part of CNN models. They involve applying a convolutional operation using learnable spatial kernels on the input. The input $\boldsymbol{h}_{k-1}$ and output $\boldsymbol{h}_k$ are feature maps of dimension $m_1^{k-1} \times m_2^{k-1} \times m_3^{k-1}$ and $m_1^k \times m_2^k \times m_3^k$. A conventional layer consists of a bank of m_1^k filters and each filter detects a particular spatial feature at every location. The i-th output feature map denoted by $\boldsymbol{h}_i^k$ is given by:

$$\boldsymbol{h}_k^i = B_i^k + \sum_{j=1}^{m_1^{k-1}} K_{ij}^k * \boldsymbol{h}_{k-1}^i \tag{2.11}$$

where $*$ is the convolution operator, B_i^k is a learnable bias matrix and K_{ij}^k is the learnable spatial kernel filter of connecting the j-th feature map of h_{k-1} with i-th feature map of $\boldsymbol{h}_k$. The success of the convolutional layer is largely due to the weight sharing strategy, where each filter is applied to the entire input, reducing the number of parameters and making the model more computationally efficient. Figure 2.4 illustrates the application of the convolutional operator to a grid-like structure, such as image pixels.

Pooling Layers. Pooling layers involves downsampling the information from the feature maps. It can be applied at any stage k of the neural network using a mean or max operator, given a spatial kernel size F^k and a stride S^k. The pooling layer takes as input a feature map of dimension $m_1^{k-1} \times m_2^{k-1} \times m_3^{k-1}$ produces a feature map of size $m_1^k \times m_2^k \times m_3^k$ with the following equation:

$$m_1^k = m_1^{k-1} \tag{2.12}$$

$$m_2^k = (m_2^{k-1} - F^k)/S^k + 1 \tag{2.13}$$

$$m_3^k = (m_3^{k-1} - F^k)/S^k + 1 \tag{2.14}$$

It is commonly used following a convolutional layer to decrease the spatial dimensions of a feature map.

Activation Functions. Activation functions (denoted by σ) introduce non-linearity into a neural network. Given an activation function σ, the output at stage k of the neural network is given by:

$$\boldsymbol{h}_k = \sigma(\boldsymbol{h}_{k-1}) \tag{2.15}$$

Empirical evidences have demonstrated that incorporating activation function helps modeling function through neural network. In practice, the Rectified Linear Unit (ReLU) operation [LeCun et al., 1989] is a common nonlinear activation function used in modern architecture.

Fully Connected Layers. are an extension of the Perceptron [Rosenblatt, 1957]. This layer applies a linear transformation on the input a vector $\boldsymbol{h}_{k-1}$ of dimension m^{k-1} for producing an output vector $\boldsymbol{h}_k$ of dimension m^k as shown below:

$$\boldsymbol{h}_k = W_k \boldsymbol{h}_{k-1} + b_k \tag{2.16}$$

where W_k is a matrix and b_k is a bias parameter. Stacking together multiple fully connected layers is known as multi-layer perceptron. Fully connected layers are usually followed by activation functions.

Recurrent Neural Network. There are other types of neural network layers proposed for tackling sequential data such as Recurrent Neural Network (RNN) [Jordan,

1990] and Long-Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997]. For such layers, the input data $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_t, ..., \mathbf{x}_T)$ is a sequence composed of T elements. RNNs use a hidden state vector, denoted by h, which is recursively updated at each timestep based on the current input. The output v is then predicted from this hidden state.

$$\boldsymbol{h}_t = \sigma_h(W_h \mathbf{x}_t + \mathcal{U}_h h_{t-1} + b_h) \tag{2.17}$$

$$\boldsymbol{v}_t = \sigma_y(W_y \boldsymbol{h}_t + b_y) \tag{2.18}$$

where σ_h and σ_y are activation functions, and W_h, W_y, $\mathcal{U}_h$ are weight matrices, while b_h and b_y are bias terms.

RNNs [Bengio et al., 1994] suffer from the vanishing and exploding gradient problem especially when dealing with long sequences. This is due to the explosion (or vanishing) of the product of derivatives during the computation of the gradient using the backpropagation through time. A common solution is to use LSTMs, which employ a gating mechanism. This mechanism allows the gradient to backpropagate more easily, essentially by smoothing out the update of the hidden vector h at each timestep by using activation functions. An alternative is the Gated Recurrent Unit (GRU), which simplifies the LSTM's gating structure while retaining similar benefits.

Multi-head Self Attention (MHSA) layers. Recently Transformers [Vaswani et al., 2017] have emerged as the core component of most of the modern DL architectures. The self-attention mechanism is fundamental unit of Transformers.

The self-attention mechanism relies on a trainable associative memory composed of (key, value) vector pairs. A query vector $q \in \mathbb{R}^d$ is matched against a set of k key vectors, which are organized into a matrix $K \in \mathbb{R}^{k \times d}$, using inner products. These inner products are then scaled and passed through a softmax function to produce k attention weights. The final attention output is computed as a weighted sum of

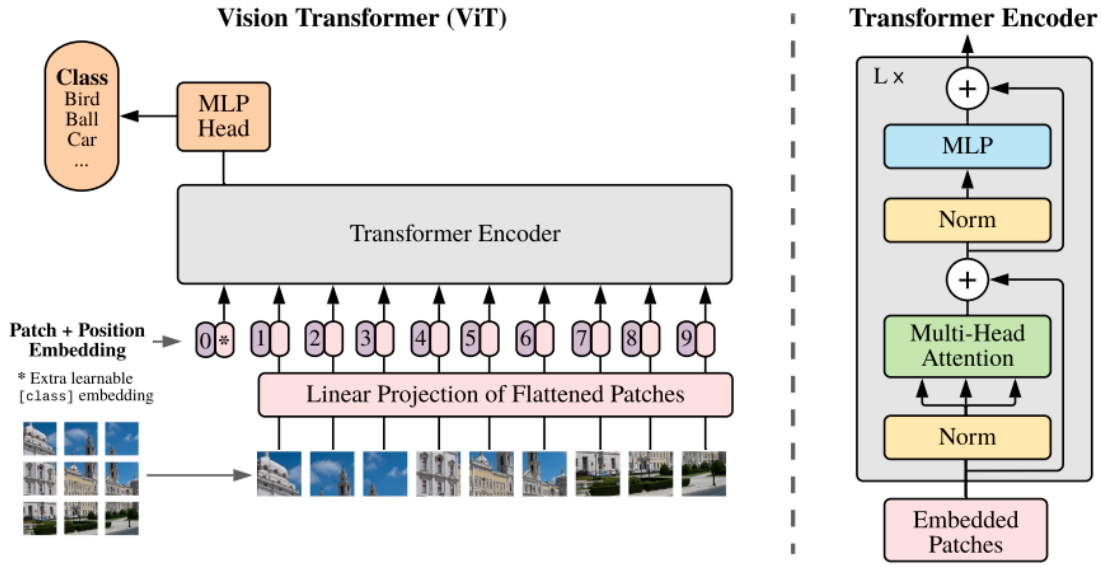


Figure 2.5: An image is divided into fixed-size patches, each of which is linearly projected. Positional embeddings are then added to these embeddings, and the resulting sequence of vectors is passed through a standard Transformer encoder. In order to perform classification, the standard approach of adding an extra learnable "classification token" to the sequence is followed. Figure from [Dosovitskiy et al., 2020].

k value vectors, packed into a matrix $V \in \mathbb{R}^{k \times d}$. When applied to a sequence of N query vectors (stacked in a matrix $Q \in \mathbb{R}^{N \times d}$), the mechanism yields an output matrix of dimensions $N \times d$ as shown below.

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^T/\sqrt{d})V, \tag{2.19}$$

where the softmax function is applied over each row of the input matrix and the d term provides appropriate normalization. Query, key and values matrices are themselves computed from a sequence of N input vectors (packed into $X \in R^{N \times D}$): $Q = XW_Q$, $K = XW_K$, $V = XW_V$, using linear transformations W_Q, W_K, W_V with the constraint $k = N$ i.e. the self-attention is in between all the input vectors.

Finally, MHSA is defined by considering h attention heads, i.e. h self-attention functions applied to the input. Each head provides a sequence of size $N \times d$. These h sequences are rearranged into a $N \times dh$ sequence that is reprojected by a linear layer into $N \times D$.

Vision Transformer. Vision Transformer (ViT) [Dosovitskiy et al., 2020] modifies the original Transformer architecture [Vaswani et al., 2017] to handle 2D image data with minimal adjustments. Specifically ViT divides an input image into N non-overlapping patches, $\mathbf{x}_i \in \mathbb{R}^{h \times w}$ (where h is the height and w is the width of the patch) and then applies a linear projection to each patch, and subsequently flattens them into 1D token embeddings $z_i \in \mathbb{R}^d$. These tokens are then arranged into a sequence that serves as the input to the Transformer encoder as given by the following equation:

$$z = [z_{cls}, \mathbf{E}\mathbf{x}_1, \mathbf{E}\mathbf{x}_2, ..., \mathbf{E}\mathbf{x}_N] + \mathbf{p}, \tag{2.20}$$

where the projection by $\mathbf{E}$ is equivalent to a 2D convolution. As shown in Figure 2.5, a learned classification token z_{cls} is prepended to this sequence, and its representation at the final layer of the encoder serves as the final representation used by the classification layer [Kenton and Toutanova, 2019].

Additionally, a learned positional embedding, $p \in \mathbb{R}^{N \times d}$, is added to the tokens to retain positional information, as the subsequent self attention operations in the transformer are permutation invariant. The tokens are then passed through an encoder consisting of a sequence of L transformer layers. Each layer comprises of MHSA [Vaswani et al., 2017], layer normalisation (LN) [Ba, 2016], and MLP blocks as follows:

$$\mathbf{y}^l = MHSA(LN(\mathbf{z}^l)) + \mathbf{z}^l \tag{2.21}$$

$$\mathbf{z}^{l+1} = MLP(LN(\mathbf{y}^l)) + \mathbf{y}^l \tag{2.22}$$

MLP consists of two linear projections separated by a GELU (Gaussian Error Linear Unit) non-linearity [Hendrycks and Gimpel, 2016] and the token-dimensionality, d, remains fixed throughout all layers. Finally, a linear classifier is employed to classify the encoded input, using either the token $z^L_{cls} \in \mathbb{R}^d$ (if it was prepended to the

input) or the global average pooling of all the tokens $\mathbf{z}^L$.

2.1.2 Unsupervised Learning

Unsupervised learning involves extracting meaningful data representations without relying on manual annotations. We review various unsupervised approaches, along with an introduction to recent advancements in self-supervised learning techniques.

Problem setup. In case of unsupervised learning, the training set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N \subset \mathcal{X}$ is composed of N training samples, where $\mathcal{X} \subset \mathbb{R}^p$ is a high-dimensional space of dimensionality p. The objective of unsupervised learning is learn a machine learning model or function f that maps an input to a representation z in a lower-dimensional space $\mathcal{Z} \subset \mathbb{R}^k$, where k is the dimensionality of the lower-dimensional space and $k \ll p$. In the $\mathcal{Z}$, the inputs with similar semantic meanings lie close to each other. Through unsupervised learning the underlying structure of $\mathcal{X}$ can be inferred.

Pretraining. Pretraining refers to the process of learning a representation $\mathcal{Z}$ in an unsupervised manner, with the goal that this learned representation will be beneficial for solving downstream tasks where only a limited number of annotated examples are available.

Clustering Methods. A standard approach to handling unannotated data involves employing clustering methods [Jolliffe, 2005, Likas et al., 2003, McLachlan and Krishnan, 2008], which aim to group similar entities together. The goal of clustering is to learn useful data representations for subsequent processing. Here we review a traditional clustering technique known as K-means.

An important design choice of a clustering method is the distance function. Typically Euclidean distance is the most commonly availed distance function as given below:

$$d(\mathbf{x}_i, \mathbf{x}_{i'}) = \sqrt{\sum_{j=1}^{p} (\mathbf{x}_{ij} - \mathbf{x}_{i'j})^2} \tag{2.23}$$

The K-means algorithm [Likas et al., 2003] applies vector quantization by assign-

ing each data point to a specific cluster among K clusters. The goal is to assign each data point to a cluster, among K clusters, which involves to learn the cluster centroids μ_k for k in $1...K$. The resulting representation z is subject to the constraint $z_k \in \{0, 1\}$ and $\sum_{k=1}^{K} z_k = 1$, ensuring a one-hot encoding of the cluster assignment of each data point. The loss function $\mathcal{L}$ consists of computing the distance between a data point $\mathbf{x}$ and its assigned centroid such as:

$$\mathcal{L}(\mathbf{x}, \mu) = \sum_{k=1}^{K} z_k d(\mathbf{x}, \mu_k) \tag{2.24}$$

where $\mu = \{\mu_1, ..., \mu_K\}$ is the set of cluster centroids. The objective function $\mathcal{J}$ to be optimized is defined as followed:

$$\mathcal{J}(\mu) = \sum_{\mathbf{x} \in \mathcal{D}} \mathcal{L}(\mathbf{x}, \mu) \tag{2.25}$$

$$\mu^* = \underset{\mu \in \Pi}{\operatorname{argmin}}\, \mathcal{J}(\theta) \tag{2.26}$$

where $\Pi = \{\mu_1, ..., \mu_K\}$. At the start of optimization, the cluster centroids are randomly initialized. The optimization process consists of two steps that are repeated until convergence of the algorithm. *First*, each data point is assigned to its nearest cluster centroid. *Second*, each cluster centroid is updated by averaging vectors assigned to the cluster.

The output representation z produced by the K-means algorithm is quite limited in its expressiveness, as it is represented by a one-hot vector. This output representation is the only part learned during optimization, since the computation of cluster centroids is solely determined by z. Furthermore, the absence of any intermediate representations can result in the loss of fine-grained information, critical for visual data. More advanced clustering techniques such as Gaussian Mixture Models [Reynolds, 2009] and spectral clustering [von Luxburg, 2007] have also been proposed. However they also do not fully resolve this limitation since they require low-dimensional input representations to function effectively.

2.1.3 Self-supervised Learning

Recently SSL, a paradigm within unsupervised learning, has gained significant attention due to its competitive performance relative to supervised models, particularly across a range of video understanding tasks. SSL exploits the underlying structure of the data, instead of relying on a manual supervisory signal (labelled training data) as used in supervised learning. SSL enables DL models to learn rich and generalizable representations by solving pretext tasks derived from unlabeled data. An example of self-supervised learning is Autoencoders, where the pretext task involves reconstructing the input.

Autoencoder. An autoencoder [Hinton and Zemel, 1993] is a type of feedforward neural network trained in an self-supervised manner to learn compact representations of input data. The model aims to encode an input $\mathbf{x}$ into a latent representation z that captures the most salient features, and then reconstruct the original input from this latent representation. To ensure that the model learns meaningful compression rather than simply memorizing the input, the dimensionality of the latent space is constrained in order to avoid convergence to a trivial identity function.

Autoencoder comprises of two networks, an encoder and a decoder. The encoder f compresses the input $\mathbf{x}$ into a low-dimensional vector representation z (the compressed or latent representation). The decoder g takes as input z and reconstructs the input $\mathbf{x}$ such that:

$$\boldsymbol{z} = f_\theta(\mathbf{x}) \tag{2.27}$$

$$\hat{\mathbf{x}} = g_\phi(\boldsymbol{z}) \tag{2.28}$$

where θ, ϕ are the learnable parameters of the encoder and the decoder respectively. The dimensionality of the latent representation z is usually smaller than the input $\mathbf{x}$ however z should be trained to be a compressed representative of $\mathbf{x}$. The loss function $\mathcal{L}$ for training the Autoencoder consists of estimating the following reconstruction

error between the input $\mathbf{x}$ and the reconstruction $\hat{\mathbf{x}}$:

$$\mathcal{L}(\mathbf{x}) = \sum_{i=1}^{N} ||\mathbf{x}_i - \hat{\mathbf{x}}_i|| \tag{2.29}$$

The loss function is optimized using SGD as explained in Section 2.1.1.

Several variants of autoencoder have been proposed to enhance their expressivity. [Vincent et al., 2008a] introduced the denoising autoencoder, a variant designed to learn robust representations by corrupting a portion of the input $\mathbf{x}$ and training the model to reconstruct the original uncorrupted input. This encourages the network to extract more meaningful and generalizable features. Similarly, contractive autoencoder [Rifai et al., 2011] was proposed to inject robustness against small perturbations in the input by introducing a regularization term $\mathcal{R}$ that enforces strong constraints on the model parameters. [Ng et al., 2011] aims to encourage sparsity in the learned latent representations by minimizing the number of active units, leveraging the Kullback–Leibler (KL) divergence as a sparsity penalty. [Kingma, 2014] extends the autoencoder framework into a generative model by adopting a variational approach. They incorporate strong assumptions on the latent variables z by using a variational approach such that the latent variable should follow a prior distribution. This encourages independence of the values of the latent representation and also leads to the learning of semantically meaningful representations.

Pre-training. [Vincent et al., 2010] demonstrate that using autoencoders for self-supervised pretraining can enhance performance on downstream tasks. Specifically, they initialize the parameters of the supervised model with those learned by the encoder during the self-supervised phase. This two-stage training strategy, where the model is first pretrained and then fine-tuned for the target task, yields improved results compared to training from randomly initialized weights.

Building on the idea of self-supervised pretraining, [Kenton and Toutanova, 2019] introduced BERT (Bidirectional Encoder Representations from Transformers) in the domain of Natural Language Processing (NLP). BERT can be interpreted as a form of denoising autoencoder [Vincent et al., 2008a]. During pretraining, a fixed

proportion (e.g., 15%) of tokens in the input sentence is masked, and the autoencoder is trained to predict the missing words based on the surrounding context. This large-scale pretraining on unannotated text corpora proves highly effective for a variety of downstream NLP tasks including sentiment analysis and text summarization.

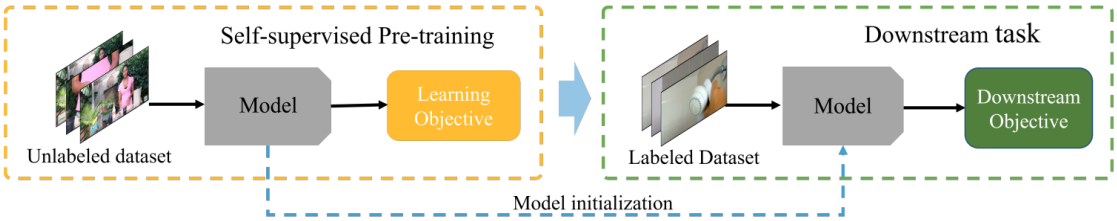


Figure 2.6: This figure illustrates the application of self-supervised pretraining to a downstream task. The process begins with pretraining a model on a large unlabeled dataset using a self-supervised objective. The resulting pretrained weights are then transferred to a model that is fine-tuned on a smaller, labeled dataset specific to the downstream task. Figure from [Schiappa et al., 2023].

Self-supervised Learning in Vision. In computer vision, several works in self-supervised learning ([Gidaris et al., 2018, Caron et al., 2018, Novotny et al., 2018]) have been proposed utilizing different pretraining strategies in order to train the encoder. Figure 2.6 illustrates the typical pretraining and finetuning pipeline used in self-supervised learning.

[Gidaris et al., 2018] propose a self-supervised approach where a CNN is trained to predict the rotation angle randomly applied to an input image. Despite its simplicity this pretext task effectively captures low-and mid-level visual features and performs comparable to supervised learning on large-scale labeled datasets such as ImageNet [Krizhevsky et al., 2012b]. [Caron et al., 2018] extend clustering-based methods to an end-to-end training paradigm by jointly learning CNN parameters and cluster assignments of the extracted features, using K-means clustering and vector quantization as supervisory signals. [Oord et al., 2018] propose to predict the future in latent space in an autoregressive manner by using a probabilistic contrastive loss. This ensures that semantic information is captured and is useful to predict the future. Following a similar strategy, [Hjelm et al., 2018] introduce Deep-InfoMax that consists of maximizing the mutual information between the input and

the output of the neural network.

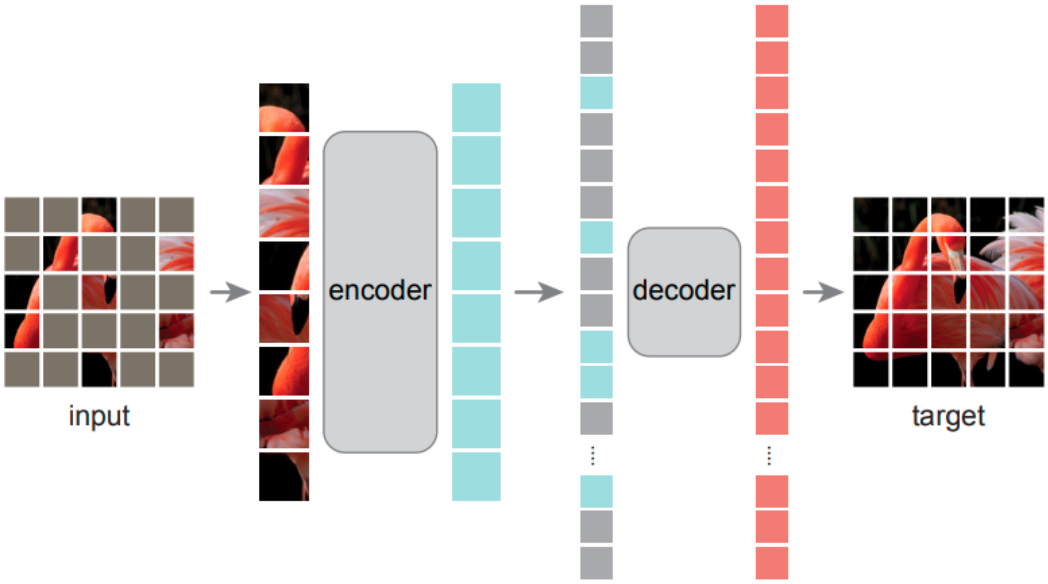


Figure 2.7: During the pretraining phase, a substantial portion of image patches (typically 75%) are randomly masked. The encoder operates only on the remaining visible patches. Following the encoder, learnable mask tokens are introduced and combined with the encoded visible representations. This combined sequence is then processed by a lightweight decoder tasked with reconstructing the original image in pixel space. Once pretraining is complete, the decoder is discarded, and the encoder is used independently on full, unmasked images for downstream recognition tasks. Figure from [He et al., 2022a]

Masked Autoencoders. Masked Autoencoder (MAE) [He et al., 2022a] performs the masking and reconstruction task with an asymmetric encoder-decoder architecture. Given an input image $I \in \mathbb{R}^{3\times H\times W}$, it is first divided into regular non-overlapping patches of size 16×16, and each patch is represented with token embedding. A subset of tokens are then randomly masked with a high masking ratio (75%), and only the remaining ones are fed into the transformer encoder ϕ_{enc}. Finally, a shallow decoder ϕ_{dec} is placed on top of the visible tokens from the encoder and learnable mask tokens to reconstruct the image. The loss function is mean squared error (MSE) loss between the normalized masked tokens and reconstructed ones in the pixel space:

$$\mathcal{L} = \frac{1}{\Omega}\sum_{p\in\Omega} |I(p) - \hat{I}(p)|^2, \tag{2.30}$$

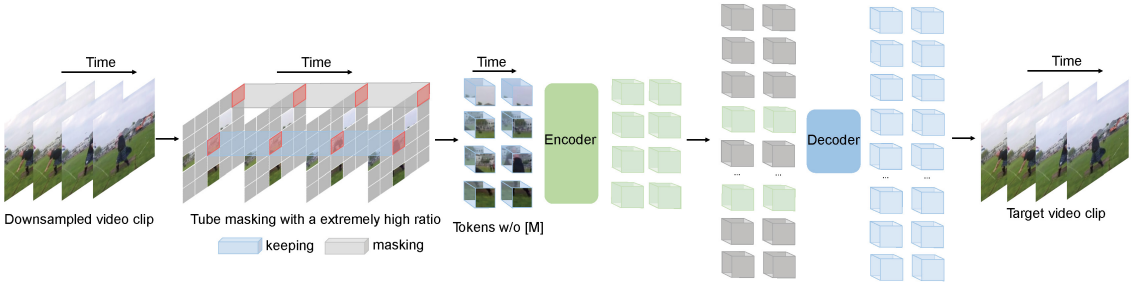


Figure 2.8: VideoMAE extends the masked autoencoding framework to video by adopting an asymmetric encoder-decoder architecture, where spatiotemporal cubes are randomly masked and subsequently reconstructed. To better leverage the high redundancy and temporal coherence inherent in video data, the model employs a tailored tube masking strategy with an exceptionally high masking ratio (ranging from 90% to 95%). This design introduces a more challenging pretraining task, thereby encouraging the model to learn more informative and robust spatiotemporal representations. Figure from [Tong et al., 2022].

where p is the token index, Ω is the set of masked tokens, I is the input image, and $\hat{I}$ is the reconstructed one. Figure 2.7 illustrates the pre-training strategy of MAE. Video Masked Autoencoders (VideoMAE) [Tong et al., 2022] (Figure 2.8) and Spatio-temporal MAE [Feichtenhofer et al., 2022] extended MAEs for video data by proposing tubelet and random space-time masking strategies.

Contrastive Learning. Another popular self-supervised learning framework is contrastive learning. Contrastive learning approaches minimize the distance between positive samples while maximizing the distance between negative samples in the joint embedding space. For vision tasks, the positives could, e.g., be random transformations of the same image (also referred as the anchor image), while the negatives are any other images. The idea of contrastive learning is not new, it can be traced back to [Chopra et al., 2005], which presented one of the earliest training objectives for deep metric learning in a contrastive fashion. However, it has been popularized recently by [Wu et al., 2018] and [Oord et al., 2018].

The loss function used in contrastive learning is derived from Noise Contrastive Estimation (NCE) [Gutmann and Hyvärinen, 2010] and its variations. The idea is to use logistic regression to discriminate the target data from noise (as the negative samples). Let $\mathbf{x}$ be the target sample $\sim P(\mathbf{x}|C=1;\theta) = p_\theta(\mathbf{x})$ and $\tilde{\mathbf{x}} \sim P(\tilde{\mathbf{x}}|C=0) = q(\tilde{\mathbf{x}})$ be the noise sample. Note that the logistic regression models the logit

(i.e. log-odds) and in this case, the goal is to model the logit of a sample $\mathbf{u}$ from the target data distribution instead of the noise distribution:

$$l_\theta(\mathbf{u}) = \log\frac{p_\theta(\mathbf{u})}{q(\mathbf{u})} = \log p_\theta(\mathbf{u}) - \log q(\mathbf{u}), \tag{2.31}$$

After converting logits ($\mathbf{u} = f_\theta(\mathbf{x})$) into probabilities with sigmoid $\sigma(.)$, the binary cross entropy loss can be applied:

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{N}\big[\log\sigma(l_\theta(\mathbf{x_i})) + \log(1 - \sigma(l_\theta(\tilde{\mathbf{x}}_\mathbf{i})))\big], \tag{2.32}$$

where $\sigma(l) = \frac{1}{1+\exp(-l)} = \frac{p_\theta}{p_\theta+q}$. In the above, the loss is applied for a single negative sample, but it can be easily extended to multiple negative samples. Based on NCE, InfoNCE uses categorical cross-entropy loss to identify the positive sample among a set of unrelated noise samples [Chen et al., 2020b]. InfoNCE is defined for $2n$ instances of images from a given n instances in a batch $\mathcal{B} = [\ t(\mathbf{x_1}), t(\mathbf{x_2}), ...t(\mathbf{x_n})]$, where $t \sim \mathcal{T}$ is a set of random transformation samples from the set of transformations $\mathcal{T}$.

$$\mathcal{L}_{\text{NCE}}(\mathbf{x_{i,j}}) = -\log\frac{\exp(\text{sim}(\mathbf{r_i}, \mathbf{r_j}))}{\Sigma_{m=0}^{2n}\mathbb{1}_{m\neq 1}\exp(\text{sim}(\mathbf{r_i}, \mathbf{r_m}))} \tag{2.33}$$

where $\text{sim}(\mathbf{x_i}, \mathbf{x_j}) = \frac{\mathbf{x}_i.\mathbf{x}_j}{||\mathbf{x}_i||||\mathbf{x}_j||}$, which is also referred to as cosine similarity and $\mathbf{r_i} = g_\phi(f_\theta(\mathbf{x_i}))$ and g_ϕ refers to MLP projection head.

Multimodal Contrastive Learning CLIP (Contrastive Language-Image Pre-training) [Radford et al., 2021], developed by OpenAI, is a vision-language pretraining framework that aligns visual and textual modalities within a shared embedding space. It achieves this by jointly training a visual encoder and a text encoder using a contrastive objective, encouraging corresponding image-text pairs to map closely together while pulling apart unrelated pairs. CLIP integrates a vision encoder model with a language encoder model. The visual component can be based on either ResNet [He et al., 2016] or Vision Transformer [Dosovitskiy et al., 2020], while the language encoder is rooted in a transformer-based model like BERT [Kenton and

Toutanova, 2019]. CLIP receives a batch of images and their corresponding text descriptions as input in each iteration. Following the encoding process, the embeddings are normalized and mapped to a joint image-text latent space. That is, the input images and texts are encoded into $I \in \mathbb{R}^{N \times D}$ and $T \in \mathbb{R}^{N \times D}$, respectively, where N denotes batch size and D represents embedding dimensionality.

Contrastive pre-training plays a crucial role in aligning image-text pairs. Diverging from conventional models that are sculpted for a singular and predefined task, CLIP's optimization revolves around contrastive pre-training between paired image-text information. In particular, N^2 image-text pairs can be constructed given a batch size of N, among which there are N matched image-text pairs and $(N^2 - N)$ unmatched image text pairs (negative pairs). The pre-training objective for the image encoder is hence denoted as:

$$\mathcal{L}_{\text{img}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\phi(I_i, T_i)/\tau)}{\Sigma_{j=1}^{N}\exp(\phi(I_i, T_j)/\tau)} \tag{2.34}$$

where $\phi(.,.)$ indicates cosine similarity, τ is a learnable temperature parameter, I_i and T_i represent the i_{th} image embedding and text embedding, respectively. The objective for the text encoder is defined symmetrically:

$$\mathcal{L}_{\text{txt}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\phi(T_i, I_i)/\tau)}{\Sigma_{j=1}^{N}\exp(\phi(T_i, I_j)/\tau)} \tag{2.35}$$

The total optimization objective of CLIP is hence calculated via the average of equation 2.34 and 2.35:

$$\mathcal{L}_{\text{total}} = \frac{\mathcal{L}_{\text{img}} + \mathcal{L}_{\text{txt}}}{2} \tag{2.36}$$

Since CLIP is pre-trained to predict whether an image matches a textual description, it naturally lends itself to zero-shot recognition. This process is accomplished by comparing image embeddings with text embeddings, which correspond to textual descriptions specifying certain classes of interest. Let I represent the image features extracted by the image encoder for a given image x, and let $\{W\}_{i=1}^{K}$ be

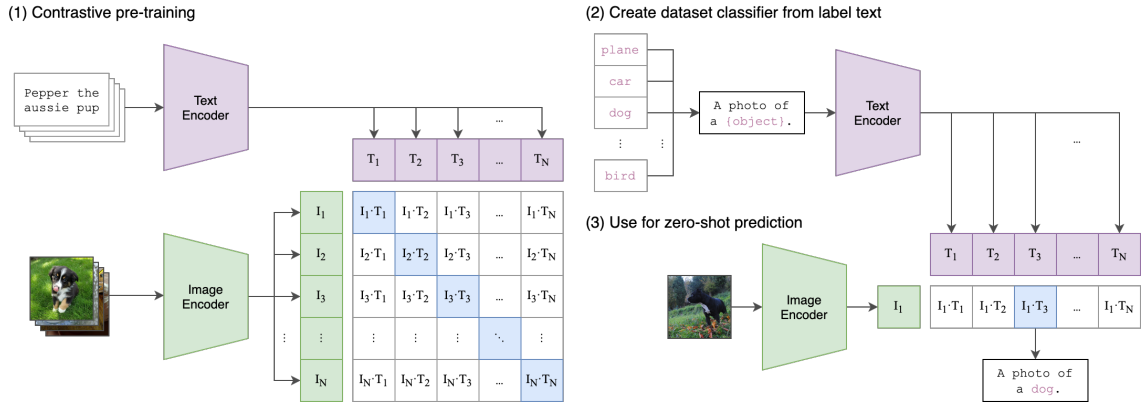


Figure 2.9: CLIP is trained by jointly optimizing an image encoder and a text encoder to correctly associate image-text pairs within each training batch. During inference, the pretrained text encoder enables zero-shot classification by encoding textual descriptions or class names from a target dataset, which are then compared to image embeddings to perform classification without additional fine-tuning. Figure from [Radford et al., 2021].

the set of class embeddings generated by the text encoder. Here, K denotes the number of classes, and each W_i is derived from a text prompt resembling "a photo of a [CLASS]", where the class token is substituted with the specific class name. The probability of prediction is then calculated as follows:

$$p(y = i/I) = \frac{\exp(\phi(I, W_i)/\tau)}{\Sigma_{j=1}^{K} \exp(\phi(I, W_j)/\tau)} \tag{2.37}$$

where τ is a temperature parameter learned during pre-training, and $\phi(.,.)$ represents the cosine similarity. In contrast with traditional classifier learning methods where closed-set visual concepts are learned from scratch, CLIP pre-training allows for the exploration of open-set visual concepts through the text encoder. This leads to a broader semantic space and, consequently, makes the learned representations more transferable to downstream tasks. The overall architecture is shown in Figure 2.9.

2.2 Supervised Models for Video Understanding

In this section, we provide a brief overview of popular supervised learning models widely used for video understanding tasks.

2.2.1 Convolutional Networks for Video Understanding

Two-Stream Convolutional Networks. [Simonyan and Zisserman, 2014] introduced the first two-stream architecture (Figure 2.10) for video-based action recognition, leveraging separate spatial and temporal pathways to extract complementary static and dynamic information. The spatial stream focuses on still frames, capturing appearance-related features such as objects and scenes as many actions are closely tied to specific objects and environments. However, for actions where appearance alone is insufficient, such as those involving objects interacting in multiple ways, the temporal stream plays a crucial role. By capturing motion patterns, it helps to resolve ambiguities and enhances the model's overall performance.

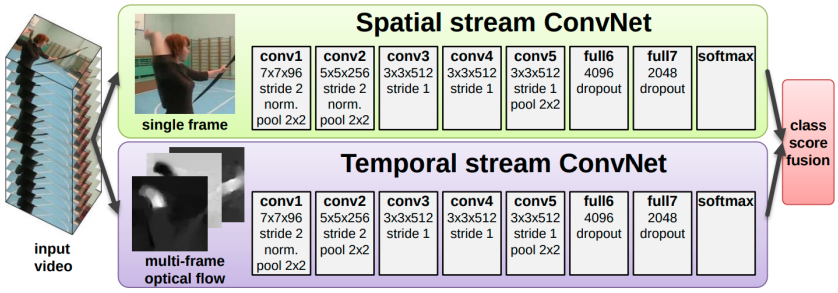


Figure 2.10: Two-Stream architecture for action recognition in video. Figure from [Simonyan and Zisserman, 2014].

During training, the spatial stream receives a single randomly sampled frame from the video, while the temporal stream receives a randomly sampled sequence of consecutive optical flow frames, capturing both horizontal and vertical motion components. Each stream is trained independently. During inference, the softmax outputs from the two CNNs are combined using the late fusion strategy to determine the action class. Two fusion strategies have been explored, one involves averaging the classification scores from both streams, and the other employs a multi-class SVM [Crammer and Singer, 2001] trained on the softmax outputs as feature vectors, with the latter showing superior performance in experiments.

The spatial network is initially pre-trained on the ImageNet dataset [Deng et al., 2009]. To fine-tune the temporal CNN on the relatively small UCF101 [Soomro et al., 2012] and HMDB51 [Kuehne et al., 2011] datasets, the authors employ a multi-task

learning approach, training the temporal stream on both datasets simultaneously. This strategy effectively increases the amount of training data, helping to mitigate overfitting.

Convolutional Two-Stream Network Fusion. Late fusion combines information from two separate CNN streams. Since the CNNs are trained independently and fusion occurs only at the classification stage, traditional two-stream architectures using this approach are unable to capture pixel-level correspondences between spatial and temporal features. To overcome this limitation, [Feichtenhofer et al., 2016] proposed a two-stream network, building on the architecture of [Simonyan and Zisserman, 2014] in which the spatial and temporal streams are integrated using a 3D convolutional layer. This allows the model to learn discriminative spatio-temporal features for the actions.

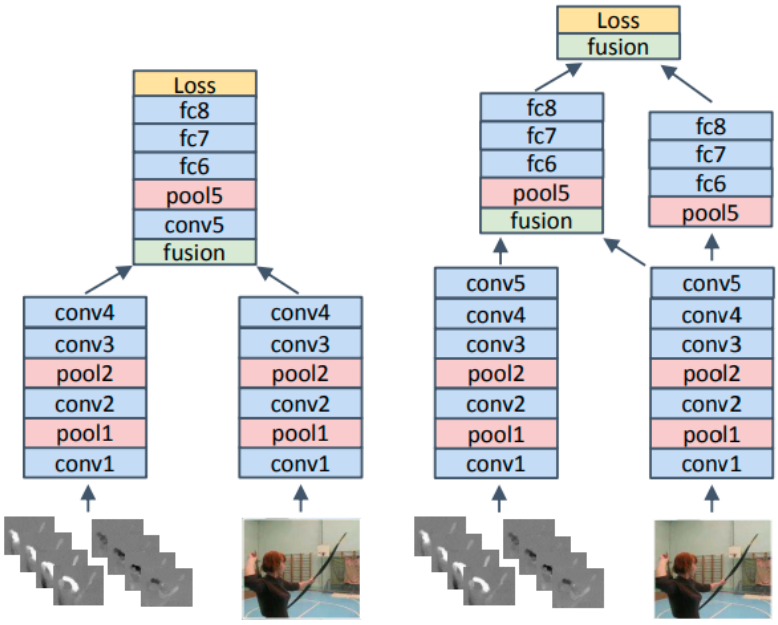


Figure 2.11: The spatial and temporal streams are fused using two different strategies. On the left, both streams are merged into a single CNN after the fourth convolutional layer. On the right, the spatial stream is integrated into the temporal stream after the fifth convolutional layer. In this configuration, the spatial CNN is preserved and later fused with the resulting spatio-temporal hybrid network. Figure from [Feichtenhofer et al., 2016].

Two different approaches were explored for fusing the spatial and temporal streams, as shown in Figure 2.11. First approach involves combining the spatial and

temporal streams into a single CNN after a certain convolutional layer (as shown in the figure on the left), which helps to reduce the overall number of parameters. Second, a dual branch architecture can be adopted. In this design, the spatial and temporal streams are processed separately through CNN layers and fused only after the fully connected layers as shown in Figure 2.11 on the left. In order to fuse the feature maps, 3D convolution is applied followed by a 3D pooling layer enabling the model to learn spatio-temporal relationships. Learning spatio-temporal relationships between the two streams is highly effective. On both UCF101 [Soomro et al., 2012] and HMDB51 [Kuehne et al., 2011], the convolutional two-stream fusion architecture outperforms other CNN-based approaches, especially those using different two-stream designs, as well as LSTM and fully 3D convolution-based models.

Temporal Modeling with 2D CNNs. [Wang et al., 2016] introduced Temporal Segment Networks (TSN), an efficient framework for capturing long-range video dynamics. Based on the observation that consecutive frames are often redundant, TSN sparsely samples frames across the video. It divides the video into n equal segments and randomly selects one frame from each segment. These sampled frames are processed individually by a frame-based CNN, and their predictions are combined using a consensus function. This approach effectively models long-term action dynamics while maintaining low computational cost.

TSN adopts the Inception network with Batch Normalization [Ioffe, 2015] as the backbone, employing a two-stream architecture with late fusion. During testing, scores from 25 uniformly sampled RGB frames and optical flow stacks are combined using a weighted average, with higher weight assigned to the spatial stream based on empirical results. Motion is encoded not only through optical flow but also via two additional modalities: RGB difference and warped optical flow. RGB difference captures pixel-wise changes between consecutive frames, providing a simple motion representation, while warped optical flow reduces camera motion to better isolate actual action dynamics. This approach achieved state-of-the-art performance on UCF101 [Soomro et al., 2012] and HMDB51 [Kuehne et al., 2011] outperforming

[Simonyan and Zisserman, 2014] on HMDB51 and UCF101, showing the efficacy of the employed sparse sampling and training techniques.

Temporal Relation Network. Temporal relation reasoning involves understanding how an entity (object or person) changes over time. [Zhou et al., 2018] noted that in many widely used datasets, such as UCF101 [Soomro et al., 2012], actions can often be recognized without explicit temporal reasoning. RGB inputs alone are typically sufficient for state-of-the-art models to perform well on such datasets. This is the case when actions are strongly characterised by the appearance of the involved objects and actors, or by the motion patterns. However, for actions that depend on temporal transformations or interactions between entities, conventional recognition methods struggle to capture the underlying dynamics effectively.

Motivated by this, [Zhou et al., 2018] proposed Temporal Relation Network (TRN). TRN was inspired by [Santoro et al., 2017], which proposed a module to learn the spatial relationship of objects in static images. TRN is effectively simple: a multi layer perceptron (MLP) θ is employed to model the relation between temporally ordered pairs of frame. More precisely, θ receives the frames' features produced by a given CNN. Another MLP ϕ operates on the output produced by θ on all the combinations of temporally ordered pairs of frames. The two MLPs are then extended to work on ordered tuples of n frames. This amounts to encoding the relationship between a sequence of frames at multiple temporal scales. The output of ϕ is used to predict the action. During training, n random ordered frames are sampled, while during testing frames are uniformly sampled throughout the video. The whole network with the TRN module is optimised with a standard cross-entropy loss.

The TRN module is integrated with the Inception network using Batch Normalization [Ioffe, 2015] and evaluated on the Something-Something [Goyal et al., 2017], 20BN-Jester [Materzynska et al., 2019], and Charades [Sigurdsson et al., 2016] datasets. TRN consistently outperforms other methods across all benchmarks datasets, highlighting the significance of temporal relation reasoning in action recog-

nition.

Temporal Shift Module. [Lin et al., 2019a] observed that although 2D CNNs are computationally efficient, they struggle to capture meaningful spatio-temporal relationships across video frames. In contrast, 3D CNNs are better at modeling temporal dynamics and yield higher accuracy, but at the expense of significantly increased computational load. To address this trade-off, [Lin et al., 2019a] introduce the *Temporal Shift Module (TSM)*, which aims to maintain high performance while minimizing computational overheads.

The TSM module enables any standard image classification CNN to function as a pseudo-3D model by introducing temporal modeling capabilities. It operates by shifting portions of the spatial feature maps across the temporal axis, as shown in Figure 2.12. Given a random set of contiguous frames and their respective feature maps, a subset of channels is shifted one frame ahead, while another subset of channels is shifted one frame behind. The remaining channels are not shifted.

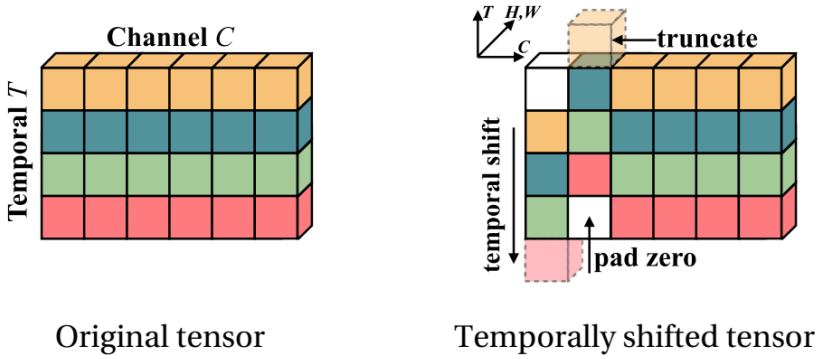


Figure 2.12: Temporal Shift Module. Spatial feature maps from four frames are stacked along the temporal dimension. The values in the first channel are shifted backward by one frame, while those in the second channel are shifted forward by one frame. The rest of the channels remain stationary. Figure from [Lin et al., 2019a].

The temporal shift mechanism interleaves spatial features from neighboring frames, allowing a 2D CNN to effectively capture spatio-temporal relationships. The approach utilizes both RGB and optical flow inputs within a late-fusion two-stream framework. During training, either 8 or 16 consecutive frames are randomly selected. At test time, the same number of frames are uniformly sampled, and their classification scores are averaged to determine the final action prediction.

TSM is not only computationally efficient, with minimal overhead from the shift operation, but also achieves high accuracy across various datasets, including Kinetics [Kay et al., 2017], Something-Something [Goyal et al., 2017], 20BN Jester [Materzynska et al., 2019], UCF101 [Soomro et al., 2012], and HMDB51 [Kuehne et al., 2011]. The proposed model is compared to a 2D baseline, specifically TSN with the same backbone CNN. TSM outperforms TSN across all datasets, with the largest improvements seen in datasets that emphasize temporal modeling. Notably, TSM achieves a 29% improvement on Something-Something and a 12% improvement on 20BN Jester compared to TSN. On Something-Something, TSM also surpasses TRN (by 7% and 8% on versions v1 and v2, respectively), which focuses on learning temporal relations, and I3D [Carreira and Zisserman, 2017b], which by employing 3D convolutions is computationally expensive.

The results show that TSM, despite using a 2D architecture, is highly effective at modeling temporal relationships. The comparison also highlights its performance relative to floating point operations (FLOPs), indicating that TSM, relying solely on cost-efficient temporal shifting, maintains low computational overhead.

Two-Stream Inflated 3D CNN. [Carreira and Zisserman, 2017a] introduced a 3D architecture in which the 2D filters from image classification CNNs are expanded to create a spatio-temporal model. This approach offers the significant advantage of leveraging successful image-based architectures, along with their pre-trained weights, for the task of video action recognition. This work was motivated by the observation that CNNs used for tasks like pose estimation and object segmentation have gained remarkable performance boost when using ImageNet pre-training.

The work comprises of experiments with several state-of-the-art architectures, as shown in Figure 2.13. The first model evaluated is a CNN+LSTM design, where visual features extracted by an image classification CNN are passed to an LSTM. This approach is commonly used to address the lack of temporal modeling in CNNs, which process only single images. While LSTMs are effective at capturing long-term, high-level dynamics, they may struggle to model salient fine-grained brief motion

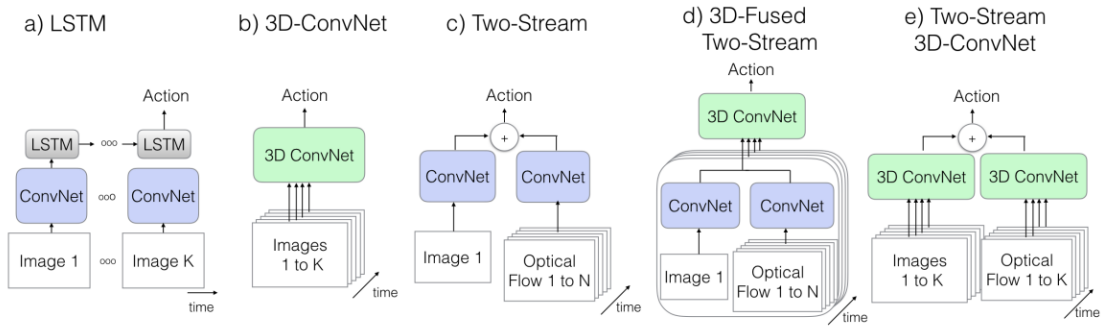


Figure 2.13: Comparison between different architectures for action recognition. Figure from [Carreira and Zisserman, 2017a].

when receiving solely spatial features.

Secondly, the work also investigates C3D (Convolutional 3D) [Tran et al., 2015]. Although 3D CNNs are naturally suited for modeling temporal information, their large number of parameters makes them challenging to train. Additionally, standard 3D CNNs like C3D cannot take advantage of pre-trained 2D models due to their architectural differences, which limits their performance as they must be trained from scratch. This major drawback was in fact the key factor that inspired the design of the inflated 3D model.

Finally, the work introduces their new model, Two-Stream Inflated 3D CNN (I3D). The core concept is that instead of developing a new 3D model from scratch, state-of-the-art 2D CNNs can be transformed into 3D models. This transformation is achieved by inflating both the convolutional and pooling kernels, essentially adding a temporal (third) dimension to the existing filters. The weights of pre-trained models are also inflated, with the parameters of the 2D convolutional filters being replicated n times to form a cube. These replicated 2D kernel weights are then averaged along the temporal axis. This inflation technique is the key contribution of I3D, giving it a significant advantage over other 3D architectures.

Experiments indicate that while 3D models are capable of capturing temporal patterns directly from RGB frames, incorporating an additional stream based on optical flow leads to improved performance. As a result, they train separate spatial and temporal CNNs and combine their predictions using late fusion during test-

ing. For training, the network processes randomly sampled stacks of 64 consecutive frames, while during testing, the entire video clip is input into the model.

The I3D model is evaluated against the aforementioned action recognition architectures on UCF101 [Soomro et al., 2012], HMDB51 [Kuehne et al., 2011], and Kinetics [Kay et al., 2017] benchmark dataset. Across all these benchmarks, I3D consistently outperforms the other methods, achieving an average improvement of 5% in top-1 accuracy. Both the late-fused and 3D-fused two-stream models show similar performance and come closest to I3D. The CNN+LSTM model, which relies solely on RGB frames, ranks slightly below the two-stream variants. C3D yields the lowest results, likely due to the absence of pre-training. Unlike the other models, which were initialized with ImageNet weights, C3D had to be trained from scratch, as pre-trained 2D weights cannot be transferred to its architecture.

Factorised 3D Convolutions. R(2+1)D [Tran et al., 2018] and S3D [Xie et al., 2018] factorise 3D convolutions into separate spatial and temporal convolution. Instead of using a 3D filter of size $t \times d \times d$, they first apply a 2D convolution of size $d \times d$ independently across t frames. This is followed by a 1D temporal convolution over the resulting t spatial feature maps using a kernel of size $t \times 1$.

This method offers two advantages. Firstly, the models are easier to optimise given that there are no 3D kernels to be tuned. Secondly, by decomposing 3D convolutions which are often more susceptible to overfitting—into separate operations, the models can achieve better classification performance. This is further validated by experiments on datasets such as Kinetics, Something-Something, UCF101, and HMDB51, where factorized 3D models matched or surpassed the accuracy of state-of-the-art methods. Notably, [Tran et al., 2018] achieved results within 1% of I3D [Carreira and Zisserman, 2017a], while [Xie et al., 2018] outperformed I3D by 3% on the Kinetics validation set.

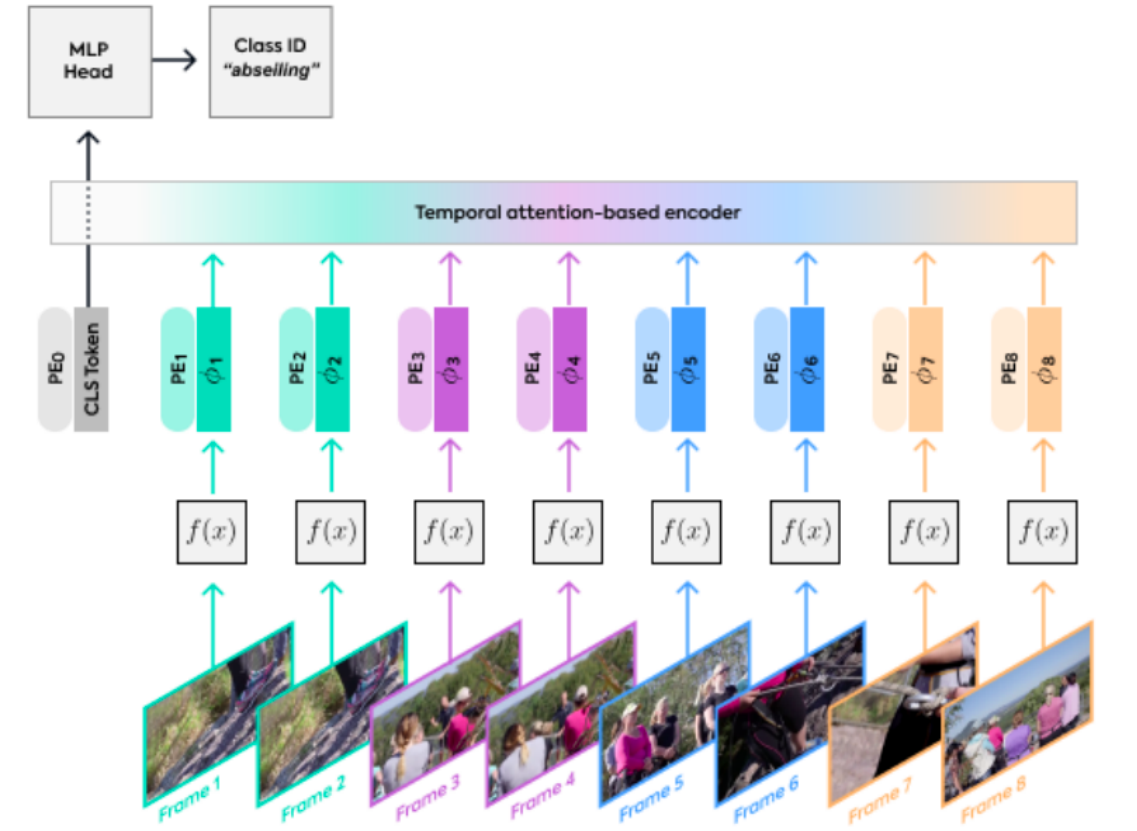


Figure 2.14: Video Transformer Network consists of a 2D spatial backbone ($f(x)$) for extracting features, followed by a temporal encoder based on attention mechanisms (Longformer [Beltagy et al., 2020]). This encoder processes the feature vectors (ϕ_i), which are enriched with positional encodings. The final class prediction is obtained by passing the [CLS] token through a classification MLP head. Figure from [Neimark et al., 2021].

2.2.2 Video Transformers for Video Understanding

Video Transformer. Based on the success of vision transformer [Dosovitskiy et al., 2020], transformer architectures have also been extended to videos such as the Video Transformer Network [Neimark et al., 2021], Video Vision Transformer (ViViT) [Arnab et al., 2021a], TimesFormer [Bertasius et al., 2021a] and Multiscale Vision Transformers (MViT) [Fan et al., 2021].

[Neimark et al., 2021] proposed the Video Transformer Network (VTN) (Figure 2.14), which first extracts frame-level features using a 2D CNN, and then employs a Transformer encoder based on Longformer [Beltagy et al., 2020] to model the temporal dependencies. Longformer is well-suited for handling long sequences

due to its linear $\mathcal{O}(n)$ complexity. The classification token output is passed through a fully connected layer to predict actions or events. Using a Transformer encoder on top of spatial features provides two key benefits. First, it enables processing an entire video in a single forward pass, and second, it enhances both training and inference efficiency by avoiding computationally expensive 3D convolutions. This design makes VTN particularly effective for analyzing long videos where interactions between entities are spread throughout the video length. The experiments on the Kinetics dataset [Kay et al., 2017] with various backbones (ResNet [He et al., 2016], ViT [Dosovitskiy et al., 2020] and DeiT [Touvron et al., 2021]) shows competitive performance.

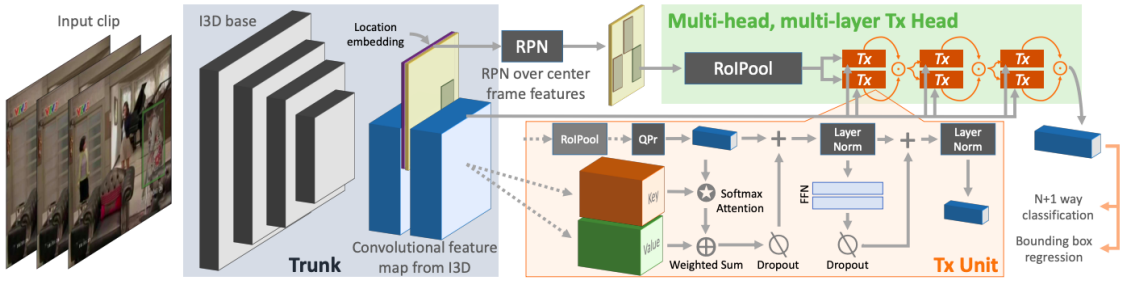


Figure 2.15: The proposed model extracts spatio-temporal features from an input video clip using the initial layers of I3D. The center frame of the feature map is passed through an RPN to generate bounding box proposals, and the feature map (padded with location embedding) and each proposal are passed through 'head' networks to obtain a feature for the proposal. This feature is then used to regress a tight bounding box and classify into action classes. The head network consists of a stack of Action Transformer (Tx) units, which generates the features to be classified. Figure from [Girdhar et al., 2019].

[Girdhar et al., 2019] proposed a variant of Transformer architecture aimed at aggregating person-specific contextual cues for action recognition and localization, as illustrated in Figure 2.15. In the beginning, the model uses a processing pipeline similar to Faster R-CNN [Ren et al., 2015b], where a backbone network extracts features that are passed to a Region Proposal Network (RPN) to generate candidate object region proposals. Region of Interest (RoI) pooling is then applied to obtain object-specific features. These features are subsequently processed by a series of MHSA layers arranged in a cascade. Within each Transformer unit, the feature

of a specific person acts as the query (Q), while features from surrounding video frames serve as the keys (K) and values (V). Positional information is explicitly embedded into the input feature map, ensuring that the self-attention mechanism is spatially aware. For a $400 \times 400 \times 64$ video clip, the key and value tensors have dimensions $16 \times 25 \times 25 \times 128$, while the query is a 128-dimensional vector. Although the model uses only RGB input, incorporating additional modalities such as optical flow or audio would significantly increase computational demands. Moreover, the Transformer was found to be less effective for precise action localization, due to its tendency to incorporate global information. Therefore, it is important to achieve the right trade-off between the global and local context for problems that demand precise delineation (e.g., action localization and segmentation).

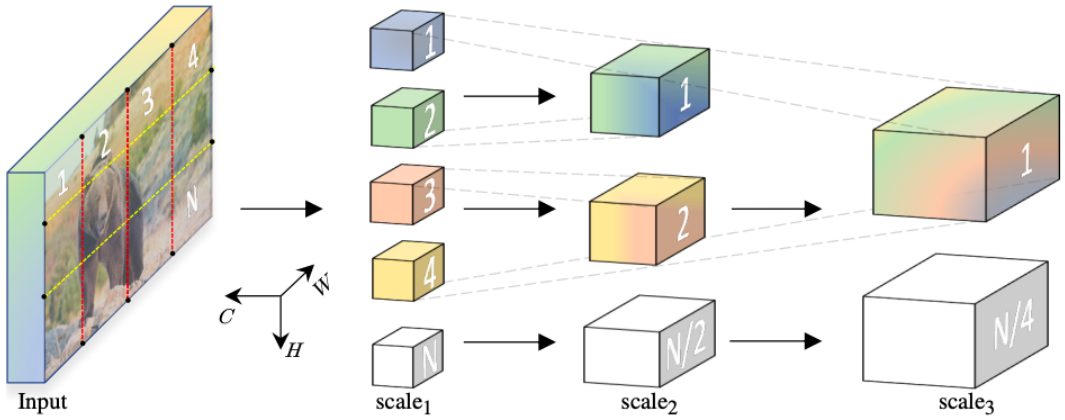


Figure 2.16: Multiscale Vision Transformers (MViT) build a hierarchical representations by transitioning from spatially dense, low-channel features to spatially coarse, high-channel ones. This is achieved through multiple stages that progressively increase the number of channels in the latent representation while reducing its length and spatial resolution. Figure from [Fan et al., 2021].

Multiscale Vision Transformer (MViT) [Fan et al., 2021] build a feature hierarchy by progressively expanding the channel capacity and reducing the spatio-temporal resolution in videos as shown in Figure 2.16.

ViViT [Arnab et al., 2021b] proposed a pure transformer for video classification problems. In particular two tokenization strategies i.e. uniform frame sampling and tubelet embedding were proposed. Several design architectures for video transformers were proposed (inspired fron ViTs) that capture pairwise interactions among all

spatio-temporal tokens. Building on this, more efficient variants were introduced by factorizing the spatial and temporal dimensions of the input video at different stages within the Transformer architecture. Figure 2.17 illustrates the four types of video transformer architecture discussed in ViViT [Arnab et al., 2021b] namely Transformer encoder, factorised encoder, factorised self-attention and factorised dot-product. These factorisations correspond to different attention patterns over space and time.

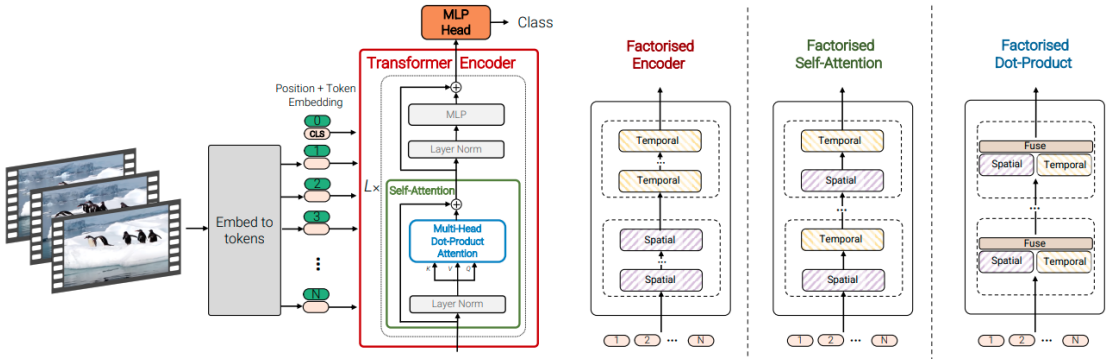


Figure 2.17: ViViT proposed a pure-transformer architecture for video classification drawing inspiration from ViT in the image domain. To efficiently handle the large number of spatio-temporal tokens, several model variants are introduced that factorise different components of the Transformer encoder across spatial and temporal dimensions. These factorisations lead to distinct attention patterns over space and time. Figure from [Arnab et al., 2021a].

TimeSformer [Bertasius et al., 2021b] also adapted the standard Transformer architecture [Vaswani et al., 2017] to video by enabling spatiotemporal feature learning directly from a sequence of frame level patches. The study compared various self-attention schemes namely space attention, joint space-time attention, divided space-time attention, sparse local global attention and axial attention for video transformer. Their findings indicate that divided space-time attention in which temporal and spatial attention are applied separately within each block—achieves the highest video classification accuracy among the evaluated design choices.

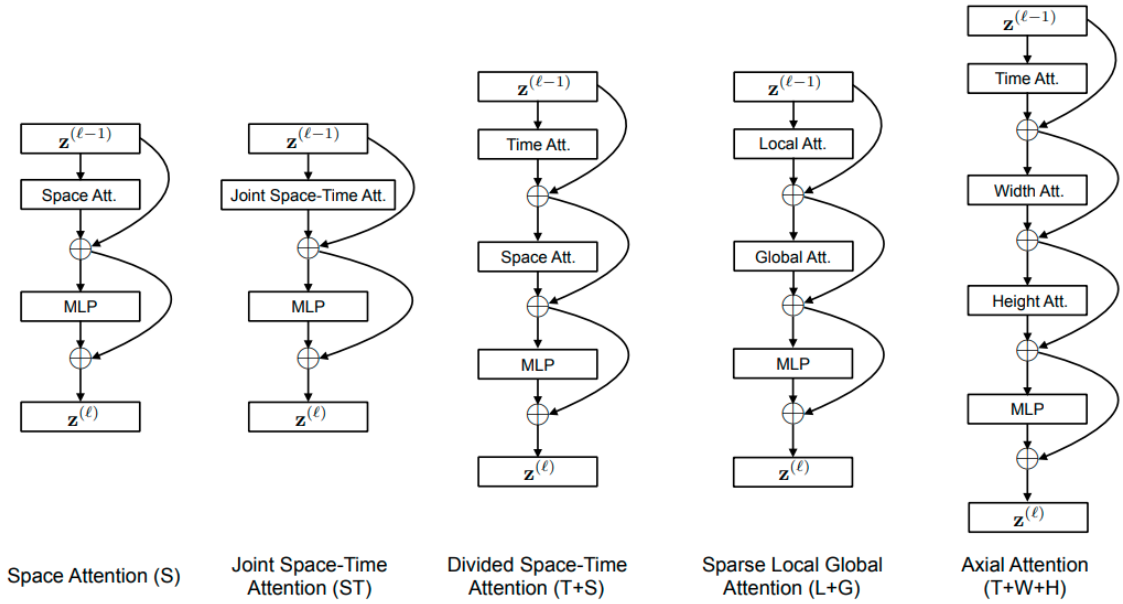


Figure 2.18: TimeSformer investigates various video self-attention blocks, where each attention layer applies self-attention [Vaswani et al., 2017] over a defined spatiotemporal neighborhood of frame-level patches. Residual connections are employed to integrate information from different attention layers within each block. Additionally, a single-hidden-layer MLP is applied at the end of each block. The complete model is built by stacking these blocks in a repeated manner. Figure from [Bertasius et al., 2021a].

2.3 Self-supervised Models for Video Understanding

Self-Supervised learning has emerged as a successful way for pre-training deep models for video understanding. It is a promising alternative where a model can be trained on large-scale datasets without the need of labels and with improved generalizability. SSL trains the model using a learning objective derived from the training samples itself. Typically, the pre-trained model is then finetuned on the target dataset as shown in Figure 2.6. In this section, we provide a brief overview of popular self-supervised learning models for video understanding tasks. We split the works in video self-supervised learning into three high level categories: pretext, generative, contrastive.

2.3.1 Pretext Task.

A pretext task refers to a SSL objective where the model is trained to solve a pre-designed task in order to learn representations that can be used later for downstream tasks. The core idea is that if a model is able to solve a complicated task that requires a high level understanding of its input, then it will learn more generalizable features. Pretext task-based methods usually depend on leveraging appearance statistics, temporal ordering, jigsaw puzzles, and playback speed information to design self-supervised training objectives.

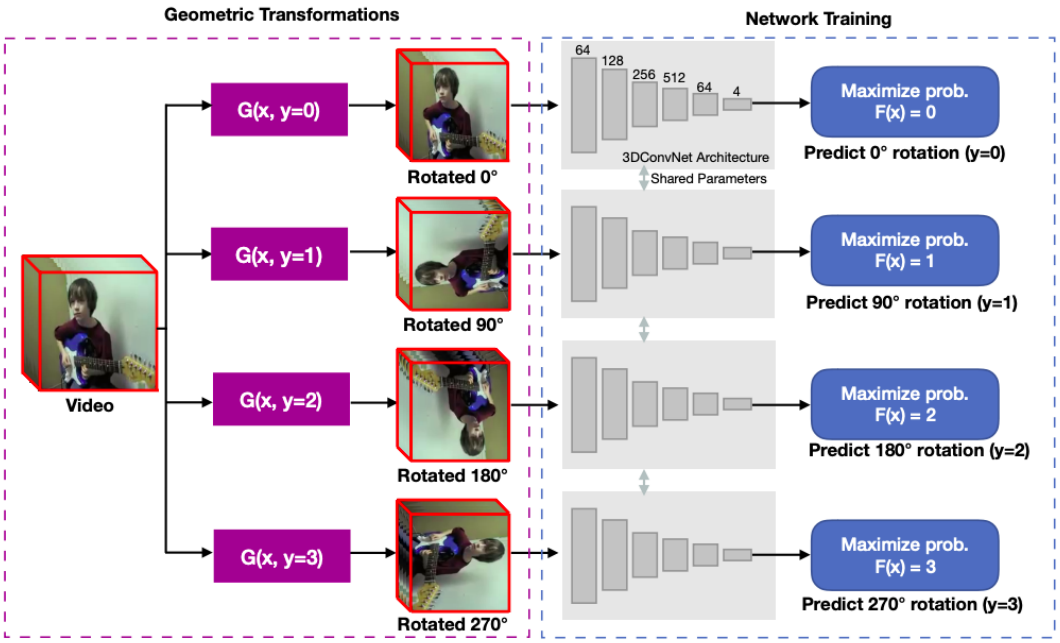


Figure 2.19: The proposed self-supervised spatiotemporal representation learning involves rotating each video by four different angles (0°, 90°, 180°, and 270°). The 3DRotNet model is then trained to predict the specific rotation applied to each input video.. Figure from [Jing et al., 2018]

.

Appearance Statistics. In this task, the model is trained to predict an appearance-transforming augmentation applied to a video clip. Typical augmentations include changes in color, rotations, and the addition of random noise. Rotation-based augmentation, initially introduced in the image domain, was extended to video in [Jing et al., 2018], where each frame of a video is individually rotated by four angles (0°, 90°, 180°, and 270°). These rotated frames are then independently fed into

a 3D CNN, and a cross-entropy loss is used to compare the predicted and actual rotations during training as shown in Figure 2.19. Although effective, this method adapts an image specific strategy to videos without explicitly modeling temporal dynamics. To better capture the temporal dimension, [Wang et al., 2019b] propose using optical flow as a pretext task. The method is based on predicting motion and spatial statistics across a vertical and horizontal grid overlaid on video frames, such as identifying where the largest motion statistic and dominant orientation statistic occurs within the grid.

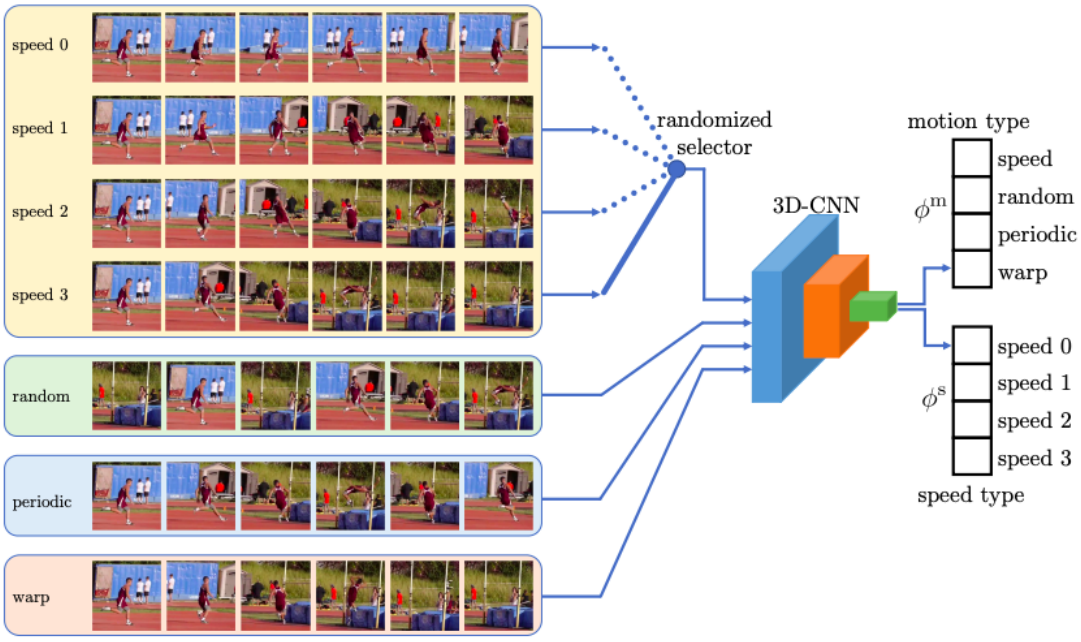


Figure 2.20: In each mini-batch, a video speed is selected from four possible choices, corresponding to different frame skipping rates in the original video. The 3D-CNN then receives a mini-batch containing a mixture of four types of transformed sequences: speed (based on the chosen frame skipping), random, periodic, and warp. The network outputs the probability of which motion type a sequence belongs to and the probability of which speed type the speed-transformed sequence has.

Playback Speed. This pretext task comprises of classifying the playback speed of an augmented video clip. Clips of t frames are extracted from each video $V \in \mathbb{R}^{T \times C \times H \times W}$, and then the playback speed is altered (either by speeding up or slowing down the video) by sampling every p-th frame, where p represents the playback rate. [Jenni et al., 2020] proposed a pretext task that adds a motion type permutation that

results in either the modified speed, random, periodic or warped transformations. The task involves two multi-class classification problems, one to identify the type of transformation applied and another to determine the playback rate p as shown in Figure 2.20. To improve performance, later approaches utilize additional tasks to the classification of p. [Yao et al., 2020] introduced a reconstruction objective, where the model encodes a modified version of a clip and reconstructs it at its original playback speed. In contrast, [Wang et al., 2020a] found that a contrastive loss yielded better results, where the original clip is the anchor, positive samples are a modified speed of the same clip, and negative samples are clips from other videos.

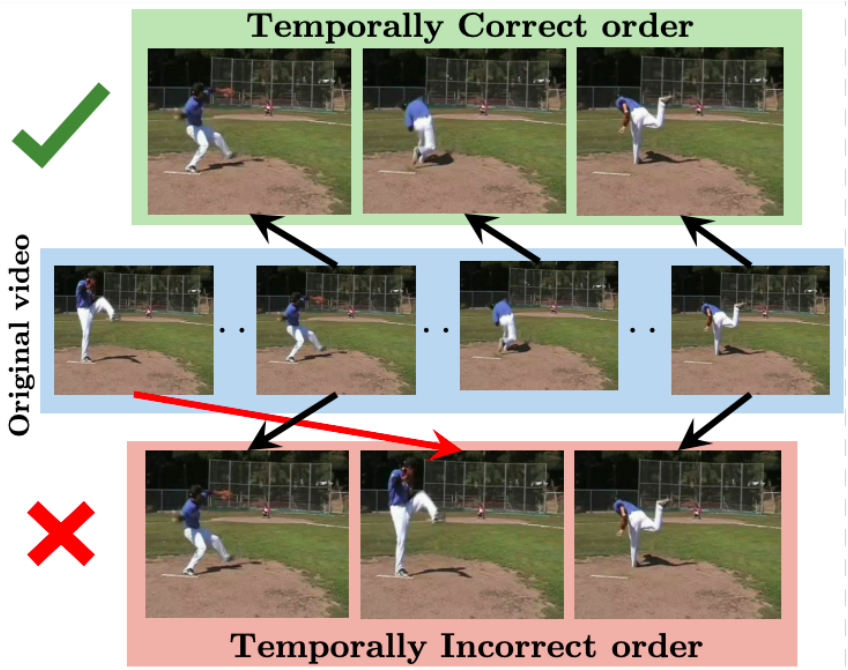


Figure 2.21: Temporal Ordering Classification Task. Figure from [Misra et al., 2016].

Temporal Ordering. In temporal order classification, each video V is divided into clips of t frames. Within each set of clips, one clip is in the correct order, while the others have their order shuffled. This pretext task called as odd-one-out learning [Fernando et al., 2017], involves training a model to determine whether a clip is in the correct or incorrect order using a binary classifier. However, in frame ordering tasks, the difference between two frames may not be sufficient to capture motion changes for certain actions. To tackle this, [Xu et al., 2019] extracts shorter sub-clips and randomizes their order. This clip-based ordering improves the

comparison because the dynamics of an action are preserved within the sub-clip. To further enhance the retention of motion dynamics, time windows are selected where the motion is most prominent, ensuring that the model can distinguish between two frames and their respective motion [Misra et al., 2016], [Lee et al., 2017]. Specifically, [Misra et al., 2016] extracts triplets of frames for each video sampled from various temporal windows, as shown in Figure 2.21. The selected frames are those that exhibit the largest motion, computed through optical flow. Negative samples consist of triplets with an incorrect order, while positive samples are correctly ordered, including reversed sequences (e.g., (t_3, t_2, t_1)). The model is trained using the odd-one-out method. In [Lee et al., 2017], the learning process is further refined by sorting frames based on pairwise feature extraction for each frame pair, instead of using the odd-one-out strategy. This method directly predicts the correct order of clips using shuffled, high-motion frames as input.

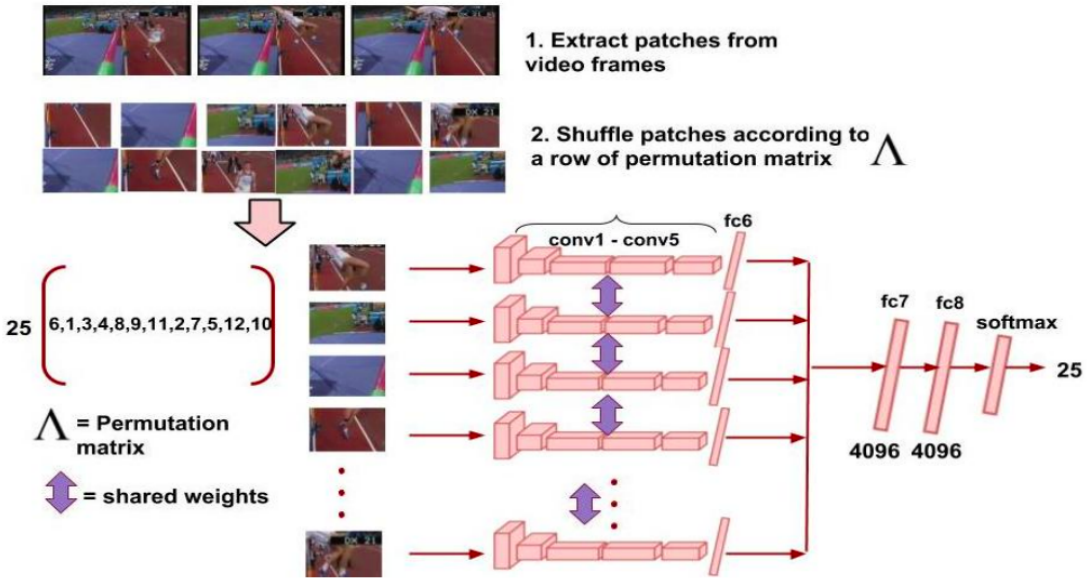


Figure 2.22: Overview of Video Jigsaw. Figure from [Ahsan et al., 2019].

Jigsaw. The jigsaw pretext task was originally introduced in the image domain by [Noroozi and Favaro, 2016], where an image is split into multiple patches and then shuffled. Each patch is assigned a number, and a permutation P is applied to reorder these numbers (e.g., 9, 4, 6, 8, 3, 2, 5, 1, 7). [Noroozi and Favaro, 2016] created a set of candidate permutations by maximizing the Hamming distance between the

original and permuted orders. The final selection of permutations consists of the top-k most dissimilar ones, forming a set S. The task is formulated as a multi-class classification problem, where the model is trained to predict which permutation $S_k \in S$ was applied to the given image.

This method has also been adapted to the video domain. A major challenge in extending the Jigsaw task to videos is the significantly larger number of patches, which leads to a drastic increase in the number of possible permutations compared to the image-based version. In [Ahsan et al., 2019] (Figure 2.22), a video is divided into clips of three frames, which are treated as a single large image, and all patches across frames are shuffled. The permutation sampling strategy was modified by constraining spatial coherence over time. The patches in a given frame were shuffled before shuffling the frames themselves. Building on the idea of combining jigsaw and frame ordering, [Zhao and Dong, 2020] addressed the increase in permutations by proposing a multi-stream CNN, where each shuffled visual sample is processed through a separate branch. To better incorporate temporal information, [Kim et al., 2019] introduced methods that represent video clips as 3D cubic representation over space and time. In particular, a video clip is divided into a 3D grid of cells, and the model is trained to classify which permutation was applied to modify the spatiotemporal structure.

2.3.2 Generative Approaches

Generative methods for representation learning leverage the capability of neural networks to synthesize videos for the purpose of pretraining. We provide a brief overview of frame prediction and masked modeling techniques within generative approaches for self-supervised video representation learning.

Frame Prediction. Instead of focusing solely on reconstructing motion, [Liang et al., 2017] and [Tian et al., 2020a] explored generating motion from RGB frames and vice versa to predict future unseen frames. They employed a combination of a discriminator and a variational autoencoder (VAE) to evaluate the quality of the

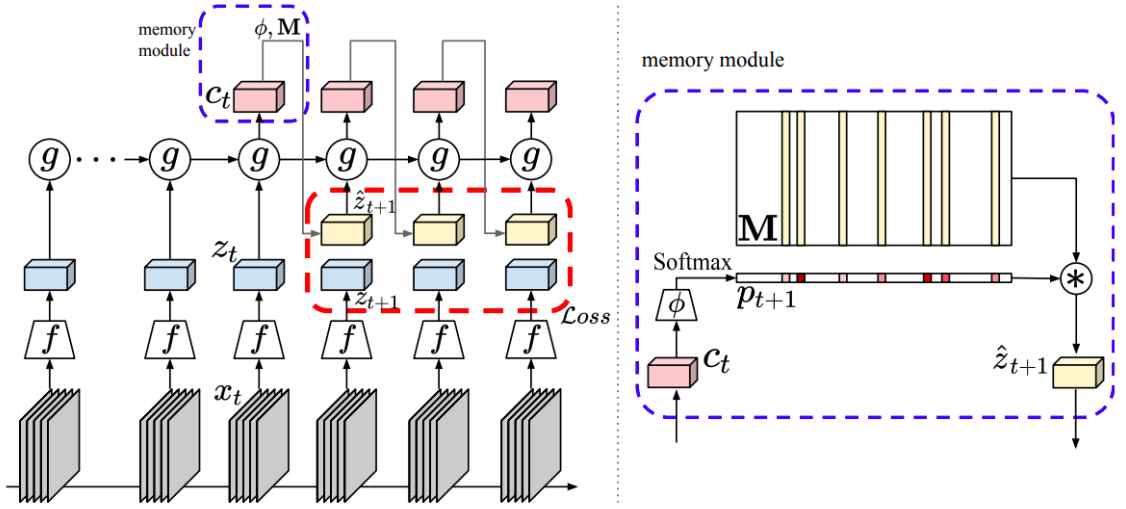


Figure 2.23: Overview of Memory Augmented Dense Predictive Coding (MemDPC). Figure from [Han et al., 2020a].

generated frames by comparing them with ground truth RGB frames and optical flow. Unlike traditional approaches that rely directly on optical flow, [Tian et al., 2020a] introduced a method that uses encoded features from pairs of frames to decode and produce low-resolution motion maps. These are then refined into high-resolution motion maps using contextual features drawn from spatial regions. The model predicts the next frame at various resolutions, with reconstruction loss used to assess the quality of each generated frame.

In order to explicitly model temporal dynamics, [Tulyakov et al., 2018] represents sequences of video frames as trajectories, allowing the model to handle videos of varying lengths. A recurrent neural network (RNN) constrains the learning of a path in the motion subspace to physically plausible motion. Recurrent structures have also been utilized in [Han et al., 2020a] which propose Memory-augmented Dense Predictive Coding (MemDPC) as shown in Figure 2.23. In this method, videos are divided into equally sized frame blocks, which are encoded into feature embeddings. These embeddings are temporally aggregated using RNNs. During training, a predictive addressing mechanism is used to access a memory bank shared between the entire dataset to draw a hypothesis and predict future blocks.

Masked Modeling. Masked Modeling was initially proposed for the image do-

main as in MAE [He et al., 2022a]. It involves masking parts of an image, where the model encodes visible patches and decodes both the visible and masked patches. This idea has also been adapted for video in several studies [Wei et al., 2022, Feichtenhofer et al., 2022, Wang et al., 2022b], with a focus on temporal components. These methods usually employ a ViT [Dosovitskiy et al., 2020] backbone to facilitate the masking task. In particular, [Feichtenhofer et al., 2022] adapts the image-based MAE framework to video by incorporating spatio-temporal learning, randomly masking space-time patches in videos and training an autoencoder to reconstruct them at the pixel level. Similarly, VideoMAE [Tong et al., 2022] applies MAE to videos using tube masking, as illustrated in Figure 2.8. BEVT [Wang et al., 2022b] extends image-based MAE by leveraging both image and video streams during the pre-training. MotionMAE [Yang et al., 2022a] focuses on temporal aspects by masking patches and feeding the encoder's output into separate Time and Space heads, which process visible and masked tokens to reconstruct frame patches and motion. MaskFeat [Wei et al., 2022] also introduces space-time cubes, where the model predicts the masked areas using context and motion cues.

2.3.3 Contrastive Learning

Contrastive learning offers a self-supervised framework that encourages representations of positive input pairs to be closer together, while pushing negative pairs further apart in the feature space. Typically, a positive pair is formed using an anchor frame and another frame from a different time point within the same video. Negative samples are generated by pairing the anchor frame with frames from different videos. The strategy used to construct positive and negative pairs is a key distinguishing aspect among various contrastive learning methods.

The most widely used contrastive learning training objectives are variations of the NCE loss. The NCE loss operates by taking a positive sample pair and a set of negative samples, with the goal of minimizing the distance between the positive pair while maximizing the distance between the negative pairs [Gutmann and Hyvärinen,

2010, Oord et al., 2018]. The primary distinction among various methods lies in how positive and negative pairs are generated to achieve this goal. In the image domain, this is typically done by applying different augmentations to an image to generate positive samples [Wu et al., 2018, Ye et al., 2019b, He et al., 2020, Chen et al., 2020b, Misra and Maaten, 2020]. These augmentations include transformations such as rotation, cropping, random grayscale, and color jittering [Ye et al., 2019b, Chen et al., 2020b].

Extending these works to video can be challenging because each video comparison adds to the memory required, especially if using multiple augmentations for multiple positive samples. Another difficulty is incorporating temporal information into the augmentations. Some approaches apply the same augmentations used in images to each individual frame [Hjelm and Bachman, 2020, Han et al., 2020b, Tian et al., 2020b, Feichtenhofer et al., 2021]. Other methods introduce additional temporal-based permutations, such as frame shuffling [Knights et al., 2021, Lorre et al., 2020]. Alternatively, some methods use motion and optical flow maps as positive samples [Rai et al., 2021b]. Furthermore, these approaches vary in how they maintain collections of negative pairs to optimize computational efficiency and memory usage, often employing techniques like memory banks or momentum encoders [Pan et al., 2021]. To enhance memory efficiency and overall performance, contrastive learning has been expanded to incorporate other video modalities, such as audio and text [Patrick et al., 2020, Amrani et al., 2021, Xu et al., 2021, Miech et al., 2019b].

Multimodal or cross-modal approaches involve learning relationships between video, audio, and/or text. Text is commonly used as a secondary modality due to the origins of NCE loss in NLP [Mnih and Kavukcuoglu, 2013] and its ability to provide rich semantic information without requiring detailed video annotations. These methods are typically trained using pairwise comparisons between separate embeddings of each modality.

2.4 Foundation Models for Video Understanding

Foundation models are large-scale, general-purpose models trained on large, diverse, unlabeled datasets using self-supervised pretraining objectives such as next-token prediction. These models are designed to learn general representations that can be adapted or fine-tuned for a wide range of downstream tasks across multiple domains (e.g., language, vision, audio, video, robotics). The field of video representation learning has undergone a rapid transformation, evolving from early transformer-based models to large-scale multimodal foundation models. Initial research in this direction led to architectures such as TimeSFormer [Bertasius et al., 2021b] and ViViT [Arnab et al., 2021b], which investigated several formulations of applying spatio-temporal attention mechanisms to the video data in supervised setting.

Such models laid the ground work for self-supervised approaches like VideoCLIP [Xu et al., 2021], which aligned video and language embeddings through contrastive learning and Video-CoCa [Yan et al., 2022], which combined causal language modeling with contrastive objectives to learn joint video-text representations enhancing the semantic understanding of video content in a self-supervised manner. VideoMAE [Tong et al., 2022] leveraged masked auto-encoding for efficient self-supervised pretraining to learn meaningful spatiotemporal features from unlabeled video data. InternVideo [Wang et al., 2022d] efficiently explored masked video modeling and video-language contrastive learning as the pretraining objectives, and selectively coordinate video representations of both of complementary frameworks in a learnable manner to boost various video applications.

The success of these vision-specific models paved the way for multimodal foundational models capable of general-purpose video-language understanding. Flamingo [Alayrac et al., 2022] demonstrated few-shot video question answering by conditioning visual-language models on sequences of video frames, setting a precedent for multimodal video understanding. Video-LLaMA [Zhang et al., 2023b] and VideoChatGPT [Maaz et al., 2024] adapted large language models to video inputs through instruction tuning and multimodal pretraining, enabling frame-wise video comprehension and

interactive dialogue-based interfaces.

Subsequent foundation models have significantly expanded in terms of scalability and generalization. LLaMA-2 [Touvron et al., 2023] and its successor LLaMA-3 [Grattafiori et al., 2024], though primarily trained on textual data, have been extended in research settings to support visual and video reasoning through adapter-based or instruction-tuned pipelines. Claude 3.5 (Anthropic, 2024) and GPT-4o (OpenAI, 2024) support limited video understanding via multi-frame input and vision adapters. NVLM [Dai et al., 2024] introduced a scalable multimodal foundation model explicitly trained on images and video, offering robust video-language alignment. Gemini 1.5 [Team et al., 2024] represents one of the most comprehensive multimodal models to date, capable of long-context reasoning over video, audio, and text. Recently released Gemini 2.5 [Comanici et al., 2025] offers unique combination of long context, multimodal and reasoning capabilities can be combined to unlock new agentic workflows. To assess and standardize these capabilities, new benchmarks and training paradigms have emerged.

2.5 Downstream Tasks

We now present an outline of three downstream tasks which can be supported by the various kinds of spatio-temporal video analysis described earlier.

Generic Event Boundary Detection. GEBD [Shou et al., 2021] focuses on localizing moments where humans naturally perceive event boundaries. As illustrated in Figure 2.24, these boundaries can occur when an action changes (e.g., from running to jumping), the subject changes (e.g., a new person appears), or the environment changes (e.g., a sudden increase in brightness), among other instances. These event boundaries are generic and taxonomy free. Spatial diversity arises from both low-level (e.g., brightness, appearance) and high-level changes (e.g., camera angle shifts, subject transitions). Temporal diversity stems from variations in action, object interactions, and speed. These spatio-temporal factors make GEBD a challenging problem.

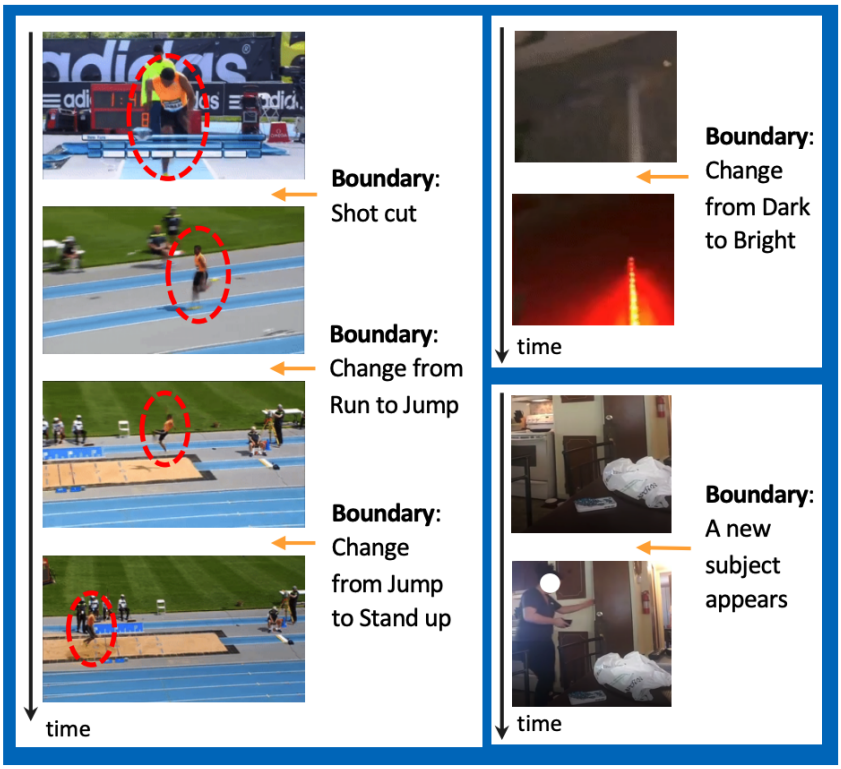


Figure 2.24: Examples of generic event boundaries include: 1) A long jump sequence segmented at a shot cut, followed by transitions between actions such as running, jumping, and standing up (with the dominant subject highlighted in a red circle). 2) A change in color or brightness. 3) New subject appears. Figure from [Shou et al., 2021].

Video Anomaly Detection. Video anomalies, as discussed in [Ramachandra et al., 2020b], can be defined as either the presence of unusual appearance or motion attributes or the occurrence of usual appearance or motion attributes in an unexpected locations or times. Figure 2.25 describes the problem of video anomaly detection.

Action Recognition. The task of action recognition [Carreira and Zisserman, 2017a, Wang et al., 2016, Simonyan and Zisserman, 2014, Varol et al., 2017, Feichtenhofer et al., 2016] involves identifying and classifying human actions or activities in videos.

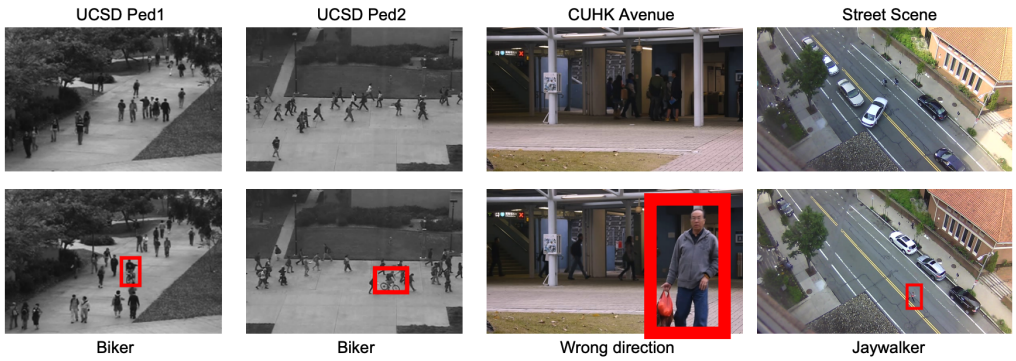


Figure 2.25: Illustration of a normal frame and an anomalous frame in single-scene benchmarks for video anomaly detection. Figure from [Ramachandra et al., 2020b].

2.6 Datasets

This subsection outlines a number of well-known open datasets which are used to compare the performance of various video analysis approaches.

Kinetics-GEBD. [Shou et al., 2021] This dataset was introduced for detecting generic event boundaries without the need of a predefined target event taxonomy. The Kinetics-GEBD training set comprises 20,000 videos randomly sampled from the Kinetics-400 training set, as described in [Kay et al., 2017]. Similarly, the Kinetics-GEBD test set includes an additional 20,000 videos randomly selected from the Kinetics-400 training set. For validation, the entire set of 20,000 videos in the Kinetics-400 validation set is used. To construct these sets, all videos in the Kinetics-400 training set are ranked by video-level class. From this ranked list, 20,000 videos are uniformly sampled to form the training set, and another 20,000 videos are sampled to establish the test set. This sampling approach ensures that the selected videos reflect a distribution similar to that of the Kinetics-400 dataset.

TAPOS. [Shao et al., 2020b] To facilitate intra- and inter-action understanding, the Temporal Action Parsing of Olympic Sports (TAPOS) dataset was constructed [Shao et al., 2020b]. TAPOS contains a total of 16,294 valid instances across 21 action classes, with an average duration of 9.4 seconds per instance. The dataset is divided into training, validation, and test sets, containing 13,094, 1,790, and 1,763 instances, respectively.

UCF Ped2. [Li et al., 2014] dataset is comprised of 16 training and 12 test videos

and all videos have the same scene in the background. The videos with normal events consist of pedestrians only, whereas the videos with anomalous events include bikes, skateboards and carts, apart from pedestrians.

Avenue. [Lu et al., 2013] This dataset is comprised of 16 training and 21 test videos with every video having the same background scene. Normal events involve people routinely walking around while the abnormal instances include abnormal objects such as bikes and abnormal human actions such as unusual walking directions, running around or throwing things.

ShanghaiTech [Luo et al., 2017c] The dataset includes 330 training and 107 test videos recorded at 13 different background locations with complex lightning conditions and camera angles, making it the one of the largest one-class anomaly detection datasets. The test split captures a total of 130 anomalous events including running, riding a bicycle and fighting.

UBnormal. [Acsintoae et al., 2022] This is a synthetic dataset with multi-scene backgrounds and a diverse set of anomalies. The dataset consists of training, validation and test split with both normal and abnormal events. The normal events include walking, talking on the phone, walking while texting, standing, sitting, yelling and talking with others. It should be noted that abnormal events in each of the train, validation and test split are different to each other. The train split includes abnormal events like *falling, dancing, walking injured, running injured, crawling, and stumbling walk*. The validation split comprises *fighting, sleeping, dancing, stealing, and rotating 360 degrees*. All the evaluations are conducted on the validation set.

Among different types of abnormal events, the train split contains *falling, dancing, walking injured, running injured, crawling and stumbling walk*. The abnormal events in the validation split include *fighting, sleeping, dancing, stealing and rotating 360 degrees*. All our evaluations in this thesis are performed on validation sets while the test split contains *running, having a seizure, laying down, shuffling, walking drunk, people and car accident, car crash, jumping, fire, smoke, jaywalking and driving outside lane* as the abnormal.

Kinetics. This is the most popular pre-training dataset for video-only self-supervised learning approaches [Kay et al., 2017]. It is a large-scale, human-action dataset containing 650,000 video clip covering either 400, 600, or 700 human action classes. Each video clip lasts around 10 seconds and is labeled with a single action class.

Something-Something v2. This dataset focuses on evaluating temporal elements in videos using everyday human activities. It is a large-scale dataset with 220,847 videos with a total of 174 activities. Some examples are "putting something into something", "turning something upside down" and "covering something with something".

UCF-101. This is one of the most popular benchmarks to evaluate models on action-recognition because of its smaller size [Soomro et al., 2012]. It has only 13,320 video clips which are classified into 101 categories. The categories fall within five types of activity: body motion, human-to-human interaction, human-to-object interaction, playing musical instruments and sports. The videos are user-generated, collected from YouTube, and have a fixed frame rate of 25 FPS and fixed resolution of 320×240.

HMDB51. This dataset contains videos collected from a variety of sources, including commercial movies and public video hosting services [Kuehne et al., 2011]. There are 6,849 video clips in total averaging 10 seconds each with 51 action categories. Each action category contains at least 101 clips. The action categories are split into five types: facial actions (e.g. smiling), face-to-object interaction (e.g. eating), general body movement, body-to-object interactions (e.g. brush hair) and human-to-human interaction (e.g. hugging).

2.7 Conclusion

This chapter presented the technical and theoretical background and basic terminology related to ML and DL in the context of video understanding required to understand the research presented in the remainder of this thesis. The following

chapters investigate the various research hypotheses and questions introduced in Chapter 1.

Chapter 3

Structured Video Representation Learning

This Chapter explores the notion of learning *structured* video representation within the SSL framework. In this Chapter, we investigate $\mathbf{H_1}$ (R_1, R_2), as introduced in Chapter 1, that designing relevant self-supervised pretext tasks can embed spatial diversity, motion-patterns, fine-grained and long-range temporal dependencies into a learned model. To validate $\mathbf{H_1}$, we develop a self-supervised approach incorporating frame-level and clip-level pretext tasks, enhance it with a differentiable motion learning module, and evaluate its performance on the GEBD task (Chapter 2). The performance achieved by our model compares favorably to the other self-supervised state-of-the-art methods. The research resulting from this work was published at the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Hawaii, 2023.

3.1 Motivation

The task of Generic Event Boundary Detection (GEBD) aims to detect moments in videos that are naturally perceived by humans as generic and taxonomy-free event boundaries. Modeling the dynamically evolving temporal and spatial changes in a video makes GEBD a difficult problem to solve. Existing approaches involve

very complex and sophisticated pipelines in terms of architectural design choices, hence creating a need for more straightforward and simplified approaches. In this chapter, we address this issue by revisiting a simple and effective self-supervised method and augment it with a differentiable motion feature learning module to tackle the spatial and temporal diversities in the GEBD task. We perform extensive experiments on the challenging Kinetics-GEBD and TAPOS datasets to demonstrate the efficacy of the proposed approach compared to other self-supervised state-of-the-art methods. We also show that this simple self-supervised approach learns motion features without any explicit motion-specific pretext task. This is important as our framework without relying on any pretext task explicitly optimizes for motion learns motion features that generalize well and incur lower computational overhead within the self-supervised learning paradigm.

Modeling videos using deep learning methods in order to learn effective global and local video representations is an extremely challenging task. Current state-of-the-art video models [Feichtenhofer et al., 2019] are built upon a limited set of predefined action classes and usually process short clips followed by a pooling operation to generate global video-level predictions. Other mainstream computer vision tasks for video processing have mainly focused on action anticipation [Miech et al., 2019a, Abu Farha et al., 2018], temporal action detection [Chao et al., 2018, Gao et al., 2017], temporal action segmentation [Lea et al., 2016a, Kuehne et al., 2014] and temporal action parsing [Pirsiavash and Ramanan, 2014, Shao et al., 2020a]. However, only limited attention has been given to understanding long form videos. Cognitive scientists [Tversky and Zacks, 2013] have observed that humans perceive videos by breaking them down into shorter temporal units, each carrying a semantic meaning and we can also reason about such long form videos. This creates an opportunity to investigate research problems to detect temporal boundaries in videos that is consistent with their semantic validity and interpretability from a cognitive point of view.

To this end, the GEBD task was recently introduced in [Shou et al., 2021][1] with an objective to study the long form video understanding problem through the lens of a human perception mechanism. GEBD aims at identifying changes in content, independent of changes in action, brightness, objects, etc., i.e. detecting generic event boundaries, making it different to tasks such as video localization [Xia and Zhan, 2020]. Video events could indicate completion of goals or sub-goals, or occasions where it becomes difficult for humans to predict what will happen next.

The recently released Kinetics-GEBD dataset [Shou et al., 2021] is the first dataset specific to the GEBD task. It is annotated by 5 different human event boundary annotators, thereby capturing the subtlety involved in human perception and making it the dataset with the greatest number of temporal boundaries (8× EPIC-Kitchen-100 [Damen et al., 2018] and 32× ActivityNet [Fabian Caba Heilbron and Niebles, 2015]). The primary challenge in the GEBD task is to effectively model generic spatial and temporal diversity as described in DDM-Net [Tang et al., 2021]. Spatial diversity is primarily the result of both low-level changes, e.g. changes in brightness or appearance, and high-level changes, e.g., changes in camera angle, or appearance and disappearance of the dominant subject. Temporal diversity, on the other hand, can be attributed to changes in action or changes by the object of interaction with different speeds and duration, depending on the subject. These spatio-temporal diversities make GEBD a difficult problem to address.

In order to address the biased nature of video models trained over predefined classes in a supervised setting, and the spatial diversity in GEBD, we leverage the power of self-supervised models. Self-supervised techniques like TCLR [Dave et al., 2022] and CCL [Kong et al., 2020] have achieved breakthrough results on various downstream tasks for video understanding. The representations learned using self-supervised learning (SSL) methods are not biased towards any predefined action class making SSL methods an ideal candidate for the GEBD task. In addition, in order to characterize temporal diversity in GEBD, learning motion information

[1]LOVEU@CVPR2021, LOVEU@CVPR2022

is essential to capture the fine-grained temporal variations that occur during the change of action scenarios. Previous methods in video modeling learn temporal motion cues by pre-computing the optical flow [Lin et al., 2018, Lin et al., 2019b, Liu et al., 2018a] between consecutive frames, which is done externally and requires substantial computation. Alternatively, methods such as those described in [Ilg et al., 2017, Fischer et al., 2015] estimate optical flow internally by learning visual correspondences between images. The motion features learnt on-the-fly can also be used for downstream applications such as action recognition as illustrated in [Zhao et al., 2018, Kwon et al., 2020].

This presents an interesting research question: how can we develop an SSL framework for video understanding that accounts for both appearance and motion features (R_1)? Do we need an explicit motion-specific training objective or can this be implicitly achieved (R_2)? We answer these questions by rethinking SSL by reformulating the training objective proposed in VCLR [Kuang et al., 2021] at clip-level and further integrating it with a differentiable motion estimation layers using the *MotionSqueeze* (MS) module introduced in [Kwon et al., 2020] to jointly learn appearance and motion features for videos. To summarise, the main contributions of this chapter are as follows:

- We revisit a simple self-supervised method VCLR [Kuang et al., 2021] and introduce a noticeable change by modifying its pretext tasks by splitting them into frame-level and clip-level to learn effective video representations (cVCLR). We further augment the encoder with a differentiable motion feature learning module for GEBD.

- We conduct extensive evaluations on the Kinetics-GEBD and TAPOS datasets and show that our approach achieves comparable performance to the self-supervised state-of-the-art methods without using enhancements like model ensembles, pseudo-labeling or the need for other modality features (e.g. audio).

- We show that a model can learn motion features under self-supervision even

without having any explicit motion specific pretext task.

3.2 Related Work

3.2.1 Generic Event Boundary Detection.

The task of GEBD [Shou et al., 2021] is similar in nature to the Temporal Action Localization (TAL) task, where the goal is to localize the start and end points of an action occurrence along with the action category. Initial attempts to address GEBD were inspired from popular TAL solvers including boundary matching networks (BMN) [Lin et al., 2019b] and BMN-StartEnd [Shou et al., 2021], which generates proposals with precise temporal boundaries along with reliable confidence scores. Shou *et al.* [Shou et al., 2021] introduced a supervised baseline Pairwise Classifier (PC), which considers GEBD as a framewise binary classification problem (boundary or not) by having a simple linear classifier that uses concatenated average features around the neighbourhood of a candidate frame. However, since GEBD is a new task, most of the current methods are an extension of state-of-the-art video understanding tasks, which overlook the subtle differentiating characteristics of GEBD. Hence there is a necessity for specialized solutions for GEBD.

DDM-Net [Tang et al., 2021] applied progressive attention on multi-level dense difference maps (DDM) to characterize motion patterns and jointly learn motion with appearance cues in a supervised setting. However, we learn generic motion features by augmenting the encoder with a MS module in a self-supervised setting. Hong *et al.* [Hong et al., 2021] used a cascaded temporal attention network for GEBD, while Rai *et al.* [Rai et al., 2021a] explored the use of spatio-temporal features using two-stream networks. Li *et al.* [Li et al., 2022b] designed an end-to-end spatial-channel compressed encoder and temporal contrastive module to determine event boundaries. Recently, SC-Transformer [Li et al., 2022a] introduced a structured partition of sequences (SPoS) mechanism to learn structured context using a transformer based architecture for GEBD and augmented it with the computa-

tion of group similarity to learn distinctive features for boundary detection. One advantage of SC-Transformer is that it is independent of video length and predicts all boundaries in a single forward pass by feeding in 100 frames, however it requires substantial memory and computational resources.

Regarding unsupervised GEBD approaches, a shot detector library[2] and PredictAbility (PA) have been investigated in [Shou et al., 2021]. The authors of UBoCo [Kang et al., 2021b, Kang et al., 2021a] proposed a novel supervised/unsupervised method that applies contrastive learning to a TSM[3] based intermediary representation of videos to learn discriminatory boundary features. UBoCo's recursive TSM[3] parsing algorithm exploits generic patterns and detects very precise boundaries. However, they pre-process all the videos in the dataset to have the same frames per second (fps) value of 24, which adds a computational overhead. Furthermore, like the SC-Transformer, UBoCo inputs the frames representing the whole video at once, whereas in our work we use raw video signals for pre-training and only the context around the candidate boundary as input to the GEBD task. TeG [Qian et al., 2021a] proposed a generic self-supervised model for video understanding for learning persistent and more fine-grained features and evaluated it on the GEBD task. The main difference between TeG and our work is that TeG uses a 3D-ResNet-50 encoder as their backbone, which makes the training computationally expensive, whereas we use a 2D-ResNet-50 model and modify it by adding temporal shift module (TSM[4]) [Lin et al., 2019a] to achieve the same effect as 3D convolution while keeping the complexity of a 2D CNN.

GEBD can be used as a preliminary step in a larger downstream application, e.g. video summarization, video captioning [Wang et al., 2022c], or ad cue-point detection [Chen et al., 2021]. It is, therefore, important that the GEBD model not add excessive computational overhead to the overall pipeline, unlike many of the examples of related work presented here.

[2] `https://github.com/Breakthrough/PySceneDetect`
[3] Temporal Self-Similarity Matrix
[4] Temporal Shift Module

3.2.2 SSL for video representation learning.

Self-supervision has become the new norm for learning representations given its ability to exploit unlabelled data [Noroozi and Favaro, 2016, Gidaris et al., 2018, Doersch et al., 2015, Asano et al., 2019, Caron et al., 2020, Zbontar et al., 2021, Bardes et al., 2021, Chen et al., 2020b, Oord et al., 2018, Krishna et al., 2021, Djilali et al., 2021]. Recent approaches devised for video understanding can be divided into two categories based on the SSL objective, namely pretext task based and contrastive learning based.

Pretext task based. The key idea here is to design a pretext task for which labels are generated in an online fashion, referred to as *pseudo labels*, without any human annotation. Examples include: predicting correct temporal order [Misra et al., 2016], Video Rot-Net [Jing et al., 2018] for video rotation prediction, clip order prediction [Xu et al., 2019], odd-one-out networks [Fernando et al., 2017], sorting sequences [Lee et al., 2017], and pace prediction [Wang et al., 2020a][5]. All these approaches exploit raw spatio-temporal signals from videos in different ways based on pretext tasks and consequently learn representations suitable for varied downstream tasks.

Contrastive learning based. Contrastive learning approaches bring semantically similar objects, clips, etc., close together in the embedding space while contrasting them with negative samples, using objectives based on some variant of Noise Contrastive Estimation (NCE) [Gutmann and Hyvärinen, 2010]. The Contrastive Predictive Coding (CPC) approach [Oord et al., 2018] for images was extended to videos in DPC [Han et al., 2019] and MemDPC [Han et al., 2020a], which augments DPC with the notion of *compressed memory*. Li *et al.* [Tao et al., 2020] extends the contrasting mutli-view framework for inter-intra style video representation, while Kong *et al.* [Kong et al., 2020] combine ideas from *cycle-consistency* with contrastive learning to propose *cycle-contrast*. Likewise, Yang *et al.* [Yang et al., 2020a] exploits visual tempo in a contrastive framework to learn spatio-temporal features.

[5]leverages contrastive learning as an additional objective as well.

Similarly, [Dave et al., 2022, Bai et al., 2020] use temporal cues with contrastive learning. VCLR [Kuang et al., 2021] formulates a video-level contrastive objective to capture global context. In the work presented in this Chapter, we exploit VCLR as our backbone objective. However, different to pretext tasks in VCLR, which perform computation only on frame level, we modify those pretext tasks to not only operate on frame-level but also on clip-level thereby leading to better modeling of the spatio-temporal features in videos. See [Schiappa et al., 2022] for a more extensive review of SSL methods for video understanding.

3.2.3 Motion estimation and learning visual correspondences for video understanding.

Motion estimation. Two-stream architectures [Feichtenhofer et al., 2016, Simonyan and Zisserman, 2014] have exhibited promising performance on the action recognition task by using pre-computed optical flow, although such approaches reduce the efficiency of video processing. Several other methods [Fan et al., 2018, Jiang et al., 2019, Lee et al., 2018a, Piergiovanni and Ryoo, 2019, Sun et al., 2018b] have proposed architectures that learn motion internally in an end-to-end fashion. The work presented in [Li et al., 2021b, Yang et al., 2020b] introduced a motion-specific contrastive learning task to learn motion features in a self-supervised setting.

Learning visual correspondences. Many recent works have proposed to learn visual correspondences between images using neural networks [Fischer et al., 2015, Han et al., 2017, Lee et al., 2019a, Min et al., 2019, Rocco et al., 2017, Sun et al., 2018a]. Regarding learning correspondences for video understanding, CPNet [Liu et al., 2019] introduced a network that learns representations of videos by mixing appearance and long-range motion features from an RGB input only. Zhao *et al.* [Zhao et al., 2018] proposed a method that learns a disentangled representation of a video, namely static appearance, apparent motion and appearance change from RGB input only. *MotionSqueeze* (MS) [Kwon et al., 2020] introduced an end-to-end trainable, model-agnostic and lightweight module to extract motion features that

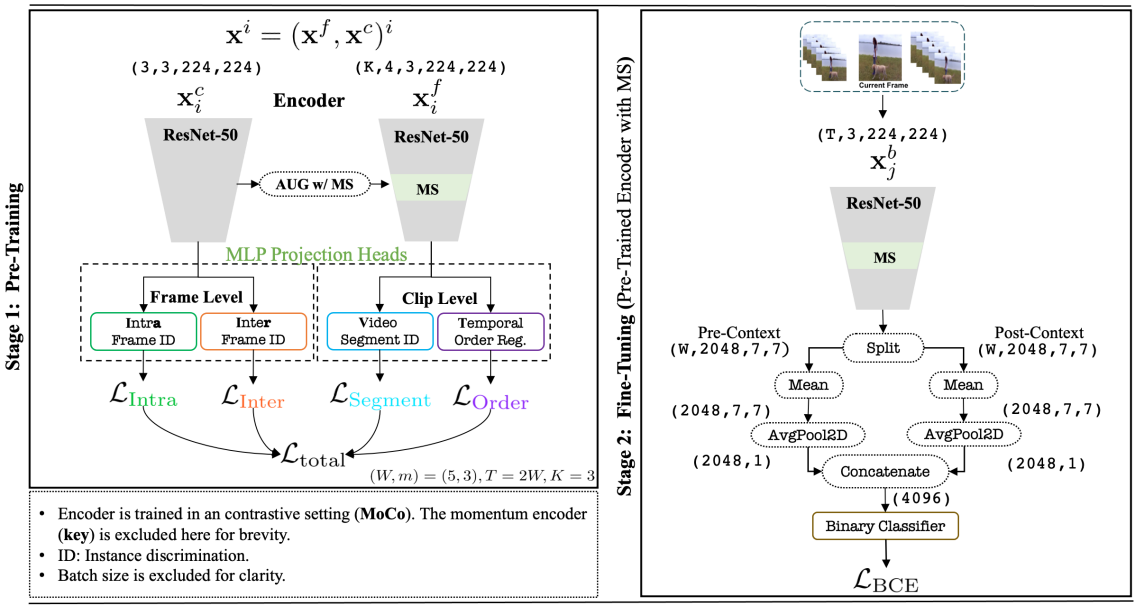


Figure 3.1: The overall architecture consists of two stages: a) **Stage 1** involves the pre-training of the modified ResNet50 encoder (augmented with a *MotionSqueeze* layer) with four pretext tasks using a contrastive learning based objective; b) **Stage 2** consists of fine tuning of the encoder on the downstream GEBD task.

does not require any correspondence supervision for learning.

3.3 Method

In order to apply a contrastive learning framework to videos specifically for generic event boundary detection, we follow the framework proposed by VCLR [Kuang et al., 2021] and make noticeable modifications to it. For simplicity, the notations are kept similar to [Kuang et al., 2021] unless otherwise explicitly stated.

3.3.1 SSL for Video Representation Learning

1: Contrastive encoder. Our processing backbone is a ResNet-50 based encoder equipped with four pretext tasks, as defined in VCLR [Kuang et al., 2021], trained following a contrastive objective as defined in MoCo-V2 [Chen et al., 2020c]. Let $\mathbf{x}_p = \mathcal{T}(\mathbf{x}_q)$ be an augmented view of an anchor image $\mathbf{x}_q$ with $\mathcal{T} \sim \mathcal{P}$ ($\mathcal{P} =$ {random scaling, color-jitter, random grayscale, random Gaussian blur, and random horizontal flip} being set of augmentations) and $\mathcal{N}^-$ negative samples. $\mathbf{x}_q$ and $\mathbf{x}_p$

are processed through query ($f_q(\mathbf{x}_q)$) and a key ($f_k(\mathbf{x}_p)$) encoder respectively. In addition, these encoders are appended with projection heads (MLP layers) to get low dimensional representations of inputs i.e. $q = g_q(f_q(\mathbf{x}_q)), p = g_k(f_k(\mathbf{x}_p))$. The overall objective can be optimized by InfoNCE loss [Oord et al., 2018]:

$$\mathcal{L}_{\text{NCE}}(q, p, \mathcal{N}^-) = -\log\left(\frac{e^{\text{sim}(q,p)}}{e^{\text{sim}(q,p)} + \sum_{j=1}^{\text{N}} e^{\text{sim}(q,n_j)}}\right), \tag{3.1}$$

where $\text{sim}(\cdot,\cdot)$ is a similarity function. We note that g_q and g_k can be thought of as a task-specific projection heads with further details below[6].

2: Pre-text setup. In order to capture different generic subtle nuances (spatial variations, temporal coherency, long range dependencies) for video understanding, the pretext tasks defined in VCLR [Kuang et al., 2021] are a good candidate for pre-training as they serve the purpose of capturing such semantics for powerful video representation from raw video signals. We alter the pretext task setup in VCLR to ensure that Intra and Inter instance discrimination (ID) tasks operate at frame-level while computation of the video segment ID and temporal order regularization tasks occurs at clip-level. Below we elaborate on the intuition behind this notion.

For frame-level pretext tasks, consider three randomly selected frames from a video, v_1, v_2 and v_3. v_1 that undergo different augmentations to generate v_1^a and v_1^+. $\mathcal{N}^-$ represents negative samples from other videos. v_1^a is processed through query encoder $f_q(v_1^a)$, while (v_1^+, v_2, v_3) is processed through key encoder $f_k(\cdot)$. While depending upon the pretext, the projection head varies across the tasks.

For clip-level pretext tasks, a video V is divided into K (set to 3) segments $\{S_1, S_2, ..., S_K\}$ of equal duration. Two tuples, comprising of 4 frame long clips, are randomly and independently sampled from each of these segments to form an anchor tuple and a positive tuple. For instance, let $c_k = \{u_1, u_2, u_3, u_4\}$ where c_k denote the ordered clip sampled from the k^{th} segment while $u_1.....u_4$ represent the frames in that clip. Similarly the anchor and positive tuple are given by $t^a = \{c_1^a, c_2^a, \dots c_K^a\}$ and $t^+ = \{c_1^+, c_2^+, \dots c_K^+\}$ respectively.

[6]subscript q, k in f and g represent the *query* and *key*

a. Intra-frame ID task. In order to model spatial diversity for the GEBD task, we adopt the intra-frame instance discrimination task proposed in VCLR [Kuang et al., 2021] to model inherent spatial changes across frames. For this task only v_1^+ is considered as a positive example while v_2 and v_3 represent negative examples. MLP heads are given by g_q^r and g_k^r while anchor embedding $q_1^a = g_q^r(f_q(v_1^a))$, positive embedding as $p_1^+ = g_k^r(f_k(v_1^+))$ and the negative sample[7] embeddings $p_2 = g_k^r(f_k(v_2))$, $p_3 = g_k^r(f_k(v_3))$. The loss objective is given by:

$$\mathcal{L}_{\text{Intra}} = \mathcal{L}_{\text{NCE}}(q_1^a, p_1^+, \{p_2, p_3\}). \tag{3.2}$$

b. Inter-frame ID task. Detecting generic event boundaries requires encoding fine-grained temporal structures from a coherent action, which are consistent with each other. To model this, inter-frame instance discrimination task considers v_1^a as an anchor frame and (v_1^+, v_2, v_3) as positive samples while $\mathcal{N}^-$ as negative samples. g_q^e and g_k^e are MLP projection heads which output the anchor embedding $q_1^a = g_q^e(f_q(v_1^a))$ and positive embeddings as $p_1^+ = g_k^e(f_k(v_1^+))$, $p_2 = g_k^e(f_k(v_2))$, $p_3 = g_k^e(f_k(v_3))$. Let $p' \in \{p_1^+, p_2, p_3\}$, hence the inter objective becomes:

$$\mathcal{L}_{\text{Inter}} = \frac{1}{3}\sum_{p'} \mathcal{L}_{\text{NCE}}(q_1^a, p', \mathcal{N}^-). \tag{3.3}$$

c. Video segment based ID task. Learning long range temporal diversity in a video is also crucial for the GEBD task. For capturing the evolving semantics in the temporal dimension we need to incorporate global video level information. The contrastive loss objective is chosen in a way that each clip in the clip anchor and clip positive tuples i.e. t^a and t^+ learn a video-level embedding through consensus operation (denoted by $\mathcal{C}$) e.g. average. Mathematically this can be represented as:

$$q_t^a = g_p^s(\mathcal{C} \mid \mathcal{C}(f_q(c_1^a)), \mathcal{C}(f_q(c_2^a)), \ldots, \mathcal{C}(f_q(c_K^a))), \tag{3.4}$$

[7] *Note*: Negative samples comes from the same video i.e. two samples as shown in Eq. (3.2).

$$p_t^+ = g_k^s(\mathcal{C} \mid \mathcal{C}(f_k(c_1^+)), \mathcal{C}(f_k(c_2^+)), \ldots, \mathcal{C}(f_k(c_K^+))), \tag{3.5}$$

$$\mathcal{L}_{\text{Segment}} = \mathcal{L}_{\text{NCE}}(q_t^a, p_t^+, \mathcal{N}^-). \tag{3.6}$$

Here, g_q^s and g_k^s represent the MLP heads, $\mathcal{C}(f_q(c_k^a))$ indicates the average over the encoder representation of the individual frame in the k^{th} clip while q_t^a, p_t^+ denotes the final embeddings for anchor and positive clip tuples. The video-level contrastive loss is given by $\mathcal{L}_{\text{Segment}}$.

d. Temporal order regularization task. In order to enforce inherent sequential structure on videos for signalling supervision in self-supervised video representation learning, we need a pretext task to learn the correct temporal order of the video data. This can also be attained through pretext tasks proposed in [Fernando et al., 2017, Wang et al., 2020a]. However, in this work we restrict ourselves to use the temporal ordering as a regularization term (denoted by $\mathcal{L}_{\text{Order}}$) within the contrastive framework as explained in Section 3.3 in [Kuang et al., 2021] though we reformulate it to include clip-level computation.

The modifications made to video segment based ID task and temporal order regularization task in VCLR to incorporate clip-level computation is referred to as `cVCLR` (clip-VCLR).

3.3.2 Motion Estimation

For learning motion features we use the *MotionSqueeze* (MS) module presented in [Kwon et al., 2020], a learnable motion feature extractor that can be inserted into any video understanding architecture to learn motion features and replace the external computation of optical flow. The motion features are learned in three steps:

1: Correlation computation. Consider $F^{(t)}$ and $F^{(t+1)}$ represent two adjacent input feature maps of spatial resolution $H \times W$ and channel dimension C. The correlation tensor $\mathbb{S}^{(t)}$ is computed by calculating a correlation score for every spatial

position $\mathtt{x}$ with respect to displacement $\mathtt{p}$ following the correlation layer implementation in FlowNet [Fischer et al., 2015]. The correlation for position $\mathtt{x}$ is only computed in neighborhood size $\mathtt{P} = 2l + 1$ by restricting a maximum displacement $\mathtt{p} \in [-l, l]^2$ and the value of $\mathtt{P}$ is set to 15.

2: Displacement estimation. The next step involves estimating the displacement map of size $H \times W \times 2$ from the correlation tensor $\mathtt{S}^{(t)}$. To get the best matching displacement for position $\mathtt{x}$, *kernal-soft-argmax* [Lee et al., 2019a] is used. In addition, a motion confidence map (of size $H \times W \times 1$) of correlation as auxiliary motion information is obtained by pooling the highest correlation on each position $\mathtt{x}$ as described in [Kwon et al., 2020]. The motion confidence map helps in identifying displacement outliers and learn informative motion features. The displacement map is then concatenated with the motion confidence map to create a displacement tensor $\mathbf{D}^{(t)}$ of size $H \times W \times 3$.

3: Feature transformation. In order to convert displacement tensor $\mathbf{D}^{(t)}$ to a relevant motion feature $\mathbf{M}^{(t)}$ (with the same channel dimension C as input $\mathbf{F}^{(t)}$), $\mathbf{D}^{(t)}$ is passed through four depth-wise separable convolutions [Howard et al., 2017] similar to [Kwon et al., 2020]. Contrary to [Kwon et al., 2020], in our work, we apply this feature transformation in a self-supervision setting to learn a displacement tensor and motion confidence map (generic motion features). Finally, the motion features $\mathbf{M}^{(t)}$ are added to the input of the next layer using an element-wise addition operation : $\mathbf{F}'^{(t)} = \mathbf{F}^{(t)} + \mathbf{M}^{(t)}$. The resulting fused feature $\mathbf{F}'^{(t)}$ is passed as input to the next layer. For more details we refer the reader to [Kwon et al., 2020].

3.3.3 Optimisation

The overall contrastive *loss objective* is given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Inter}} + \mathcal{L}_{\text{Intra}} + \mathcal{L}_{\text{Segment}} + \mathcal{L}_{\text{Order}}. \tag{3.7}$$

Our encoder is augmented with an MS module (introduced after `conv3_x`[8]) to jointly learn appearance and motion features. More precisely, a ResNet-50 [He et al., 2016] model is adopted as the CNN encoder and we insert a TSM[4] [Lin et al., 2019a] for each residual block of ResNet. Each of the four losses contribute equally to $\mathcal{L}_{\text{total}}$ although weighing them appropriately might boost performance. The overall framework is illustrated in Figure 3.1.

3.3.4 Architectural Design Choice

Temporal Shift Module (TSM) [Lin et al., 2019a] is inserted in every residual block of the ResNet50 encoder. A *MotionSqueeze* module is added after the `conv3_x` layer of the ResNet50 encoder.

Table 3.1: Modified ResNet50 Encoder

Layers	ResNet-50	Modified ResNet-50	Output size
conv1	$7 \times 7, 64$, stride 2		112×112
conv2_x	3×3, max-pool, stride 2		56×56
	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{TSM} \\ 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	
conv3_x	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} \text{TSM} \\ 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 128 \end{bmatrix} \times 4$	28×28
MS Module	✗	✓	28×28
conv4_x	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{TSM} \\ 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	14×14
conv5_x	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{TSM} \\ 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	7×7

It should be noted that our encoder definition is consistent with the architecture design introduced in ResNet [He et al., 2016] and is different from the encoder in the work of *MotionSqueeze* in [Kwon et al., 2020] as shown in Table 3.1.

Table 3.2: F1 scores on the Kinetics-GEBD validation set with Relative Distance threshold ranging from 0.05 to 0.5 with step of 0.05. ‡: soft-labels, †: hard-labels. * is pretrained on Kinetics-400 [Kay et al., 2017] dataset.

	Rel. Dis Threshold	Finetuning	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	avg
Supervised	BMN† [Lin et al., 2019b]	✗	0.186	0.204	0.213	0.220	0.226	0.230	0.233	0.237	0.239	0.241	0.223
	BMN-SE [Lin et al., 2019b]†	✗	0.491	0.589	0.627	0.648	0.660	0.668	0.674	0.678	0.681	0.683	0.640
	TCN-TAPOS [Lea et al., 2016b]†	✓	0.464	0.560	0.602	0.628	0.645	0.659	0.669	0.676	0.682	0.687	0.627
	TCN [Lea et al., 2016b]†	✗	0.588	0.657	0.679	0.691	0.698	0.703	0.706	0.708	0.710	0.712	0.685
	PC [Shou et al., 2021]†	✓	0.625	0.758	0.804	0.829	0.844	0.853	0.859	0.864	0.867	0.870	0.817
	SBoCo-Res50 [Kang et al., 2021b]†	✗	0.732	-	-	-	-	-	-	-	-	-	0.866
	SBoCo-TSN [Kang et al., 2021b]†,*	✗	0.787	-	-	-	-	-	-	-	-	-	0.892
	DDM-Net [Tang et al., 2021]†	✗	0.764	0.843	0.866	0.880	0.887	0.892	0.895	0.898	0.900	0.902	0.873
	Li et.al. [Li et al., 2022b]‡	✗	0.743	0.830	0.857	0.872	0.880	0.886	0.890	0.893	0.896	0.898	0.865
	SC-Transformer [Li et al., 2022a]‡	✗	**0.777**	**0.849**	**0.873**	**0.886**	**0.895**	**0.900**	**0.904**	**0.907**	**0.909**	**0.911**	**0.881**
Un-supervised	SceneDetect [Catellano, 2014]	✗	0.275	0.300	0.312	0.319	0.324	0.327	0.330	0.332	0.334	0.335	0.318
	PA - Random [Shou et al., 2021]	✗	0.336	0.435	0.484	0.512	0.529	0.541	0.548	0.554	0.558	0.561	0.506
	PA [Shou et al., 2021]	✗	0.396	0.488	0.520	0.534	0.544	0.550	0.555	0.558	0.561	0.564	0.527
	UBoCo-Res50 [Kang et al., 2021b]	✗	0.703	-	-	-	-	-	-	-	-	-	0.866
	UBoCo-TSN [Kang et al., 2021b]*	✗	0.702	-	-	-	-	-	-	-	-	-	0.892
(Self-supervised)	TeG-PS [Qian et al., 2021a]†	✓	0.699	-	-	-	-	-	-	-	-	-	-
	TeG-FG [Qian et al., 2021a]†	✓	0.714	-	-	-	-	-	-	-	-	-	-
	Ours†	✓	0.680	0.779	0.806	0.818	0.825	0.830	0.834	0.837	0.839	0.841	0.809
	Ours‡	✓	0.711	0.777	0.791	0.795	0.798	0.799	0.801	0.802	0.802	0.803	0.788

3.4 Experimental Setup

3.4.1 Implementation Details.

Stage 1: Pre-training. We closely follow VCLR [Kuang et al., 2021] to train the encoder. The model was pre-trained end-to-end with the objective as defined in Eq. (3.7) on 2 NVIDIA GeForce RTX-2080Ti GPUs with an effective batch size ($\mathcal{B}$) of 8 distributed across the GPUs (4 each) with temperature set to 0.01 across all pretext tasks. The input to the frame level and clip level pretext task is $(\mathcal{B}, 3, 3, 224, 224)$ and $(\mathcal{B}, K, 4, 3, 224, 224)$ respectively with $K = 3$. TSM[4] and *MotionSqueeze* is only applied on clip level task. The encoder is initialised to MoCo-v2 [Chen et al., 2020c] weights with negative samples $\mathcal{N}^-$ (queue size) set to 8192 and is trained with SGD for 400 epochs with a warm-start of 5 epochs following a cosine decay with base learning rate of 0.01. *Pre-training* is only performed on the Kinetics-GEBD [Shou et al., 2021] dataset.

Stage 2: Finetuning. Input to the encoder is based on the temporal window (W=5) which defines a context over a candidate frame (before and after) with a stride $m = 3$[9] resulting into a 4D tensor $(10, 3, 224, 224)$ as input. (W, m) can be thought of as hyper-parameter, setting a larger value of each might introduce

[8]notation as in [He et al., 2016]

[9]selecting one frame out of every 3 consecutive frames

Table 3.3: F1 scores on the TAPOS validation set with Relative Distance threshold ranging from 0.05 to 0.5 with step of 0.05. ‡: soft-labels, †: hard-labels. (-) : Not clear.

	Rel. Dis Threshold	Finetuning	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	avg
Supervised	ISBA [Ding and Xu, 2018]	-	0.106	0.170	0.227	0.265	0.298	0.326	0.348	0.369	0.376	0.384	0.314
	TCN [Lea et al., 2016b]	✗	0.237	0.312	0.331	0.339	0.342	0.344	0.347	0.348	0.348	0.348	0.330
	CTM [Huang et al., 2016]	-	0.244	0.312	0.336	0.351	0.361	0.369	0.374	0.381	0.383	0.385	0.350
	Transparser [Shao et al., 2020a]	-	0.289	0.381	0.435	0.475	0.500	0.514	0.527	0.534	0.540	0.545	0.474
	PC [Shou et al., 2021]†	✓	0.522	0.595	0.628	0.647	0.660	0.666	0.672	0.676	0.680	0.684	0.643
	DDM-Net [Tang et al., 2021]†	✗	0.604	0.681	0.715	0.735	0.747	0.753	0.757	0.760	0.763	0.767	0.728
	SC-Transformer [Li et al., 2022a]‡	✗	**0.618**	**0.694**	**0.728**	**0.749**	**0.761**	**0.767**	**0.771**	**0.774**	**0.777**	**0.780**	**0.742**
Un-supervised	SceneDetect [Catellano, 2014]	✗	0.035	0.045	0.047	0.051	0.053	0.054	0.055	0.056	0.057	0.058	0.051
	PA - Random [Shou et al., 2021]	✗	0.158	0.233	0.273	0.310	0.331	0.347	0.357	0.369	0.376	0.384	0.314
	PA [Shou et al., 2021]	✗	0.360	0.459	0.507	0.543	0.567	0.579	0.592	0.601	0.609	0.615	0.543
(Self-supervised)	Ours†	✓	0.573	0.614	0.639	0.656	0.669	0.679	0.687	0.693	0.700	0.704	0.661
	Ours‡	✓	0.586	0.624	0.648	0.663	0.675	0.685	0.692	0.697	0.704	0.708	0.668

noise information when two different boundaries lie close to each other, a smaller value might be unable to capture the necessary context information for a boundary. Among the 5 annotations available for Kinetics-GEBD for every video, the ones with highest annotator F1 consistency score is used for fine-tuning. We fine-tune the model end-to-end with a binary cross entropy (BCE) (boundary is 0/1) as the objective augmented with Gaussian smoothing ($\sigma = 3$) for soft labeling as in [Li et al., 2022a]. The learning rate set to $7.5e^{-4}$ for Kinetics-GEBD, while for TAPOS it was set to $1e^{-4}$. Balance sampling is applied to each batch during training to avoid class imbalance. We finetune the model for 8 epochs and use early stopping to find the best model.

To select the final boundary predictions for the video, we apply post-processing scheme on the obtained boundary proposals. *First*, proposals should be greater then a threshold of 0.5. *Second*, we aggregate all the proposals within a 1 second time window. The code for reproducing the results presented in this Chapter is available at `https://github.com/rayush7/motion_ssl_gebd`.

3.4.2 Evaluation Protocol.

We conduct evaluation on two datasets Kinetics-GEBD [Shou et al., 2021] and TAPOS [Shao et al., 2020a]. For evaluation, we follow the standard evaluation protocol explained in [Shou et al., 2021], which uses the F1 score as the measurement metric. *Rel. Dis* (Relative Distance) is used to decide whether a detected

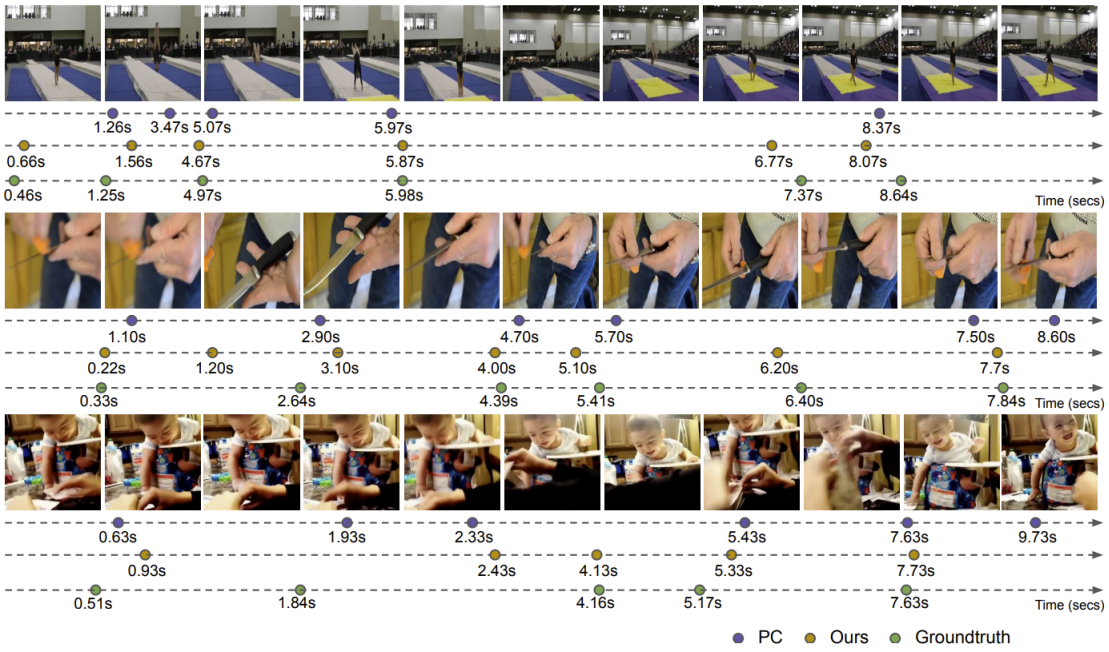


Figure 3.2: Qualitative Analysis I: visualization of some detected boundaries on the validation set of Kinetics-GEBD. Compared with baseline PC [Shou et al., 2021], our method produces more precise boundaries that are consistent with the ground truth.

event boundary is correct (if detection probability ≥ 0.5) or otherwise incorrect. More formally, *Rel. Dis* is defined as the error between detected and ground-truth timestamps, divided by the length of the whole video. F1 score calculated at *Rel. Dis* threshold 0.05 was used as the evaluation metric for the GEBD challenge[10]. We compare our detection results with all annotations (5 annotations per video for Kinetics-GEBD and 1 annotation for TAPOS) in the same video and select the annotation with the highest F1 score.

3.4.3 Results

We perform extensive quantitative and qualitative studies on the given datasets. In Tables 3.2 and 3.3, we report F1 scores for different thresholds ranging from 0.05 to 0.5 with a step of 0.05, for the Kinetics-GEBD and TAPOS datasets respectively. On the Kinetics-GEBD dataset, our model outperforms the supervised baseline PC [Shou et al., 2021] with *Rel. Dis* threshold 0.05 and is also comparable with other

[10]LOVEU@CVPR2021, LOVEU@CVPR2022

Table 3.4: Ablation study on validation set of TAPOS and Kinetics-GEBD for F1 score at *Rel. Dis* threshold 0.05

Method	TAPOS	Kinetics-GEBD
Vanilla VCLR	0.496	0.596
cVCLR	0.502 (↑)	0.605 (↑)
+ *MotionSqueeze*	0.573 (↑)	0.680 (↑)
+ Soft labels	0.586 (↑)	0.711 (↑)

state-of-the-art unsupervised/self-supervised GEBD models like UBoCo [Kang et al., 2021b] and TeG [Qian et al., 2021a] in terms of performance. Table 3.2 illustrates the result on the Kinetics-GEBD dataset. On the TAPOS dataset, which consists of Olympic sport videos with 21 action classes, we have a similar observation. Our model outperforms the supervised baseline PC [Shou et al., 2021] at the *Rel. Dis* threshold of 0.05 and is comparable on other thresholds. Other state-of-the-art methods on TAPOS like DDM-Net [Tang et al., 2021] and SC-Transformer [Li et al., 2022a] fall in the supervised category and cannot be directly compared with our results. We were unable to find other state-of-the-art un/self-supervised models for GEBD to directly compare our results with, on the TAPOS dataset shown in Table 3.3.

We also perform a qualitative analysis of boundaries detected by our method and compare them with the supervised baseline PC [Shou et al., 2021] and the ground truth annotation in Figure 3.2. Figure 3.3 shows a visualization of the motion confidence map learned by the MS module during *pre-training*. We observe that motion confidence generalizes well to the TAPOS dataset too, which was not used for pre-training. This validates that the MS module learns general motion features even in a self-supervised setting without any explicit motion specific pretext task. In addition, our model's event boundary detection results on the TAPOS dataset further justifies that our model is a generic event boundary detector, as after fine-tuning it generalizes beyond Kinetics-GEBD dataset to the TAPOS benchmark. We also found that linear evaluation (freezing the encoder) on the downstream GEBD task resulted in poor performance.

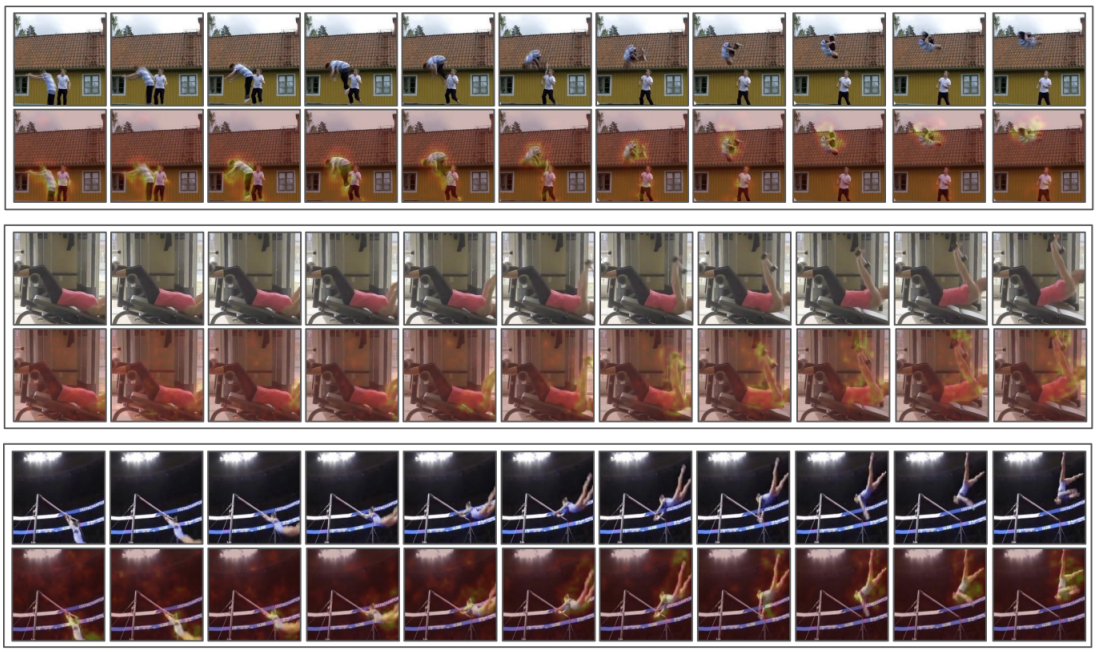


Figure 3.3: Qualitative Analysis II: visualization of the learned motion confidence map. The **first two** blocks (categories: jumping on trampoline and situp respectively) are taken from the Kinetics-GEBD dataset, while the **bottom** block (category: uneven bar) is derived from TAPOS. In each block, the first row shows the RGB frames while the second depicts the motion confidence map learnt by the model. **Note**: the model is only pre-trained on Kinetics-GEBD but it generalizes to the TAPOS dataset as well.

3.4.4 Ablation Studies

1: Does *MotionSqueeze* through self-supervision helps? As shown in Figure 3.3, the MS module shows high confidence in regions of images that are more dynamic. The module learns optical flow in an online fashion even under self-supervision. Intuitively, temporal order regularization and video segment instance discrimination pre-text tasks implicitly complements the MS module to learn generic motion features. From Table 3.4, we observe that by incorporating the MS module, the F1@0.05 score on the GEBD task increases by 7.5% on Kinetics-GEBD and 7.1% on TAPOS, which is a significant increase.

2: Does soft labelling helps in boosting the performance? Kinetics-GEBD has 5 annotators to capture human perception differences but this introduces ambiguity. Ideally the neighbouring frames of the candidate boundary frame should also have a high value of the ground truth label. To tackle this issue, we use Gaussian smoothing ($\sigma = 3$) to create soft labels from hard labels, which ensures that model avoids making over confident predictions for the event boundary. As shown in Table 3.4, soft labels improves the F1@0.05 score by 3.1% on Kinetics-GEBD and 1.3% on TAPOS dataset.

3.5 Conclusions and Discussion

This Chapter addresses the hypothesis $\mathbf{H_1}$ (Chapter 1) which deals with the notion of embedding a structure into the learned video representation by designing relevant self-supervised pretext tasks. In particular, the goal of this chapter is to evaluate $\mathbf{R_1}$: *How can we leverage the power of SSL to capture spatio-temporal diversities and relationships involved in videos?* and $\mathbf{R_2}$: *How can we develop an SSL framework for video understanding that accounts for both appearance and motion features? Do we need an explicit motion-specific training objective, or can this be implicitly achieved?* The downstream task selected for this study is the task of GEBD (Chapter 2,3).

To summarise in this Chapter, we presented a self-supervised model that can

be pre-trained for the GEBD task. The GEBD task is an ideal problem for self-supervised learning given that the task aims to learn generic boundaries and is not biased towards any predefined action categories from pre-trained state-of-the-art action recognition models. In order to learn spatial and temporal diversity we reformulate SSL objective at frame-level and clip-level to learn effective video representations (cVCLR) (**answering $\mathbf{R_1}$**). In addition we augment our encoder with a MS module and find this indeed compliments the overall performance on the downstream GEBD task. Furthermore, the motion features learnt are generic since the model is only pre-trained on Kinetics-GEBD but generalizes to TAPOS dataset as well (**answering $\mathbf{R_2}$**). Through our extensive evaluation, we achieve comparable performance to self-supervised state-of-the-art methods on the Kinetics-GEBD as shown in Table 3.2.

However, there are limitations with the work presented in this Chapter. First, we have not used more powerful models, e.g. transformers as in [Li et al., 2022a], or cascaded networks as in [Hong et al., 2021]. Second, since MS module is directly applied on feature maps, it learns global motion features. However, in GEBD the boundaries are generic and every type of motion may not indicate a boundary, hence a more fine-grained motion module can boost the performance. Third, due to computational constraints, our self-supervised model is only pre-trained on the Kinetics-GEBD dataset; however, pre-training the model on Kinetics-400 could yield even better performance on the downstream GEBD task.

Chapter 4

Robust Video Representation Learning

In this chapter, we examine how to embed *robustness* into learned video representations in a self-supervised setting. To this end, we investigate $\mathbf{H_2}$ (R_3, R_4), as introduced in Chapter 1, which posits that robustness to spatio-temporal perturbations can be achieved by exposing the model to near-distribution (PAs) samples during SSL pre-training while retaining its sensitivity to real-world anomalies. To verify $\mathbf{H_2}$, we investigate the VAD task within the OCC setting. From the available normal data, we generate near-distribution samples or PAs, using generative models such as diffusion models [Rombach et al., 2022] or availing of mixup augmentation [Zhang et al., 2018] to distort the optical flow [Zach et al., 2007] and use them for pre-training the VAD framework. The spatial, temporal and semantic information extracted from PAs also facilitate the aggregation of several anomaly indicators, which can further help in detecting real-world anomalies. The anomaly detection performance achieved by our model is competitive against the other state of-the-art reconstruction based methods. The research resulting from this work was published at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, Seattle, 2024.

4.1 Motivation

Video Anomaly Detection (VAD) is an open-set recognition task, which is usually formulated as a one-class classification (OCC) problem, where training data is comprised of videos with normal instances while test data contains both normal and anomalous instances. Recent works have investigated the creation of pseudo-anomalies (PAs) using only the normal data and making strong assumptions about real-world anomalies with regards to the abnormality of objects and their speed in order to inject prior information about anomalies in an autoencoder (AE) based reconstruction model during training. This Chapter proposes a novel method for generating generic spatio-temporal PAs by inpainting a masked-out region of an image using a pre-trained Latent Diffusion Model and further perturbing the optical flow using mixup to emulate spatio-temporal distortions in the data. In addition, we present a simple unified framework to detect real-world anomalies under the OCC setting by learning three types of anomaly indicators, namely reconstruction quality, temporal irregularity and semantic inconsistency. Extensive experiments on four VAD benchmark datasets, namely Ped2, Avenue, ShanghaiTech and UBnormal, demonstrate the effectiveness of our work against other existing state-of-the-art PAs generation and reconstruction-based methods under the OCC setting. Our analysis also examines the transferability and generalisation of PAs across these datasets, offering valuable insights by identifying real-world anomalies through PAs.

Video Anomaly Detection [Liu et al., 2018a, Liu et al., 2021, Ionescu et al., 2019a, Zaheer et al., 2020a, Gong et al., 2019a, Park et al., 2020, Astrid et al., 2021a, Astrid et al., 2021b, Sultani et al., 2018, Pourreza et al., 2021, Georgescu et al., 2021b, Georgescu et al., 2021a, Ji et al., 2020, Wang et al., 2022a, Zaheer et al., 2022b] refers to the task of discovering the unexpected occurrence of events that are distinct and follow a deviation from known normal patterns. The rarity of anomalies in the real-world and the unbounded nature (open-set recognition [Geng et al., 2020]) of their diversities and complexities have led to unbalanced training datasets for VAD, making it an extremely challenging task. Therefore VAD is commonly

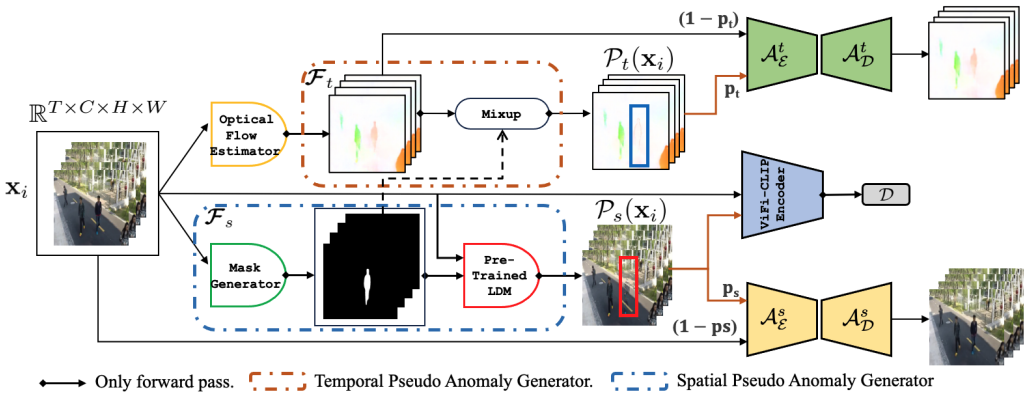


Figure 4.1: The overall architecture of our approach consists of spatio-temporal PAs generators. Spatial PAs generator (eq. 4.2) : $\mathcal{F}_s(\text{stack}(\mathbf{x}, \mathbf{x}\odot\mathbf{m}, \mathbf{m}); \theta)$and temporal PAs (eq. 4.3) : $\mathcal{F}_t(\phi(\mathbf{x_t}, \mathbf{x_{(t+1)}}))$. The spatial and temporal PAs are sampled with probabilities p_s and p_t, respectively. Our VAD framework unifies estimation of reconstruction quality (eq. 4.4), temporal irregularity (eq. 4.5) and semantic inconsistency.

addressed as an OCC problem where only normal data is available to train a model [Hasan et al., 2016, Zhao et al., 2017a, Luo et al., 2017d, Luo et al., 2017a, Gong et al., 2019a, Park et al., 2020, Astrid et al., 2021a, Astrid et al., 2021b, Georgescu et al., 2021a, Liu et al., 2021].

Reconstruction-based approaches exploiting an AE are usually adopted to tackle the OCC task [Astrid et al., 2021a, Astrid et al., 2021b, Park et al., 2020, Gong et al., 2019a]. The intuition behind this is that during training, the AE would learn to encode normal instances in its feature space with the assumption that during the test phase a high reconstruction error would correspond to an anomaly and a low reconstruction error would indicate normal behaviour. Contrary to this, [Gong et al., 2019a, Astrid et al., 2021a, Zaheer et al., 2020a] observed that when trained in this setting, the AE learns to reconstruct anomalies with high accuracy, resulting in a low reconstruction error in the testing phase. Hence, the capability of the AE to distinguish normal and anomalous instances is greatly diminished (Figure 1a in [Astrid et al., 2021a]).

[Park et al., 2020, Gong et al., 2019a] introduced a memory-based AE to restrict the reconstruction capability of the AE by recording prototypical normal patterns during training in the latent space therefore shrinking the capability of the AE to

reconstruct anomalous data. However, such methods are highly sensitive to memory size. A small-sized memory may hinder reconstruction of normal data as memorising normal patterns can be interpreted as severely limiting the reconstruction boundary of the AE, resulting in failure to reconstruct even the normal events during the testing phase (Figure 1b in [Astrid et al., 2021a]).

Astrid *et al.* [Astrid et al., 2021a] proposed the generation of two types of PAs (patch based and skip-frame based) to synthetically simulate pseudo-anomalous data from normal data and further introduced a novel training objective for the AE to force the reconstruction of only normal data even if the input samples are anomalous. Patch based PAs are generated by inserting a patch of a specific size and orientation from an intruder dataset (e.g. CIFAR-100) using the SmoothMixS [Lee et al., 2020a] data augmentation method while in order to create skip-frame based PAs, a sequence of frames is sampled with irregular strides to create anomalous movements in the sequence. The intuition behind this training procedure is based on limiting the reconstruction boundary of the AE near the boundaries of the normal data resulting in more distinctive features between normal and anomalous data (Figure 1c in [Astrid et al., 2021a]). A notable limitation of the approach proposed in Astrid *et al.* [Astrid et al., 2021a] is its heavy reliance on a predefined set of assumptions and inductive biases. These assumptions encompass various aspects, including the specific intruding dataset selected for patch insertion, the patch's size and orientation, and the idea that altering the movement speed by skipping frames could introduce temporal irregularities into the normal data.

With such assumptions, there is no guarantee that the test anomalies which comprise of an unbounded set of possible anomalous scenarios would comply with pseudo-anomalous samples. This creates a need for more generic solutions for creating PAs from the normal data. Since VAD is an open-set recognition problem and anomalies present an inexhaustible set of possibilities, every pseudo-anomaly synthesiser carries strong or weak inductive biases and thus it is inherently challenging to emulate real-world anomalies through PAs. Furthermore, there are other

challenges, such as the fact that certain normal behaviours are rare but possible and therefore not well represented in the normal data. This presents an interesting research question: *"Is it possible to synthetically generate generic PAs by introducing spatio-temporal distortions into normal data in order to detect real-world anomalies effectively?, and importantly, can such PAs transfer across multiple VAD datasets?"*

Our work is motivated by [Astrid et al., 2021a] and extends it by addressing its drawbacks and proposing a more generic PAs generator. We focus on generating PAs by injecting two different types of anomaly indicators, the first being distortion added through image inpainting performed by a pre-trained latent diffusion model (LDM) [Rombach et al., 2022], the second being the addition of temporal irregularity through perturbation of the optical flow [Zach et al., 2007] using mixup [Zhang et al., 2018]. Our simple VAD pipeline focuses on reconstructing the spatio-temporal PAs and also measures the semantic inconsistency between normal samples and PAs using semantically rich ViFi-CLIP [Rasheed et al., 2023] features. This *unifies estimation of reconstruction quality, temporal irregularity and semantic inconsistency* under one framework. We conduct an extensive study on understanding the generalisation and transferability of such PAs over real-world anomalies. Overall, our main contributions are:

- We propose a novel and generic spatio-temporal pseudo-anomaly generator for VAD encompassing inpainting of a masked out region in frames using an LDM and applying mixup augmentation to distort the optical flow.

- We introduce a simple unified VAD framework that measures and aggregates three different indicators of anomalous behaviour, namely reconstruction quality, temporal irregularity and semantic inconsistency in an OCC setting.

- Extensive experiments on *Ped2, Avenue, ShanghaiTech* and *UBnormal* show that our method though not objectively state-of-the-art (SOTA) achieves comparable performance to other existing SOTA PAs generation and reconstruction based methods under the OCC setting (Table 4.4, 4.3) without any end-

to-end finetuning or any post-processing. This validates our hypothesis that our method is a generic video anomaly detector and our spatio-temporal PAs generation process is transferable across multiple datasets.

4.2 Related Work

4.2.1 Restricting Reconstruction Capacity of an AE

A standard approach to address VAD is to adopt an OCC strategy by training an AE model to reconstruct the input data [Hasan et al., 2016, Zhao et al., 2017a, Luo et al., 2017d, Luo et al., 2017a, Gong et al., 2019a, Park et al., 2020, Astrid et al., 2021a]. During training, only normal inputs are used for learning the AE with the assumption that reconstruction of anomalies during testing would yield a higher reconstruction error. However, in practice it has been shown that the AE can also reconstruct anomalous data [Gong et al., 2019a, Astrid et al., 2021a, Zaheer et al., 2020a]. [Gong et al., 2019a, Park et al., 2020] mitigated this issue by augmenting the AE with memory-based techniques in the latent space to restrict the reconstruction capability of an AE. However the performance of such methods are directly impacted by the choice of the memory size, which may over-constrain the reconstruction power of the AE resulting in poor reconstruction of even the normal events during testing.

To alleviate this issue, [Astrid et al., 2021a, Astrid et al., 2021b] utilised data-heuristic based PAs built on strong assumptions to limit the reconstruction capacity of the AE. Patch-based PAs were generated by inserting a patch from an intruding dataset (CIFAR-100) into the normal data by using techniques such as SmoothMixS [Lee et al., 2020a]. For modeling motion-specific anomalous events, PAs were generated by skipping frames with different strides to induce temporal irregularity. The training configuration was set up to minimise the reconstruction loss of the AE with respect to the normal data only. PAs can be interpreted as a type of data-augmentation [Bengio et al., 2011, Krizhevsky et al., 2012b], where instead of creating more data of the same distribution, pseudo-anomalous data is created

that belongs to a near-distribution i.e. between the normal and anomaly distributions. [Tang et al., 2020b, Zhang et al., 2019] adopted adversarial training to generate augmented inputs, which were also effective as an adversarial example for the model.

Our method falls into the category of restricting the reconstruction capability of an AE. Inspired by the method introduced in [Astrid et al., 2021a], we propose a novel technique for simulation of generic spatio-temporal PAs without making bold assumptions about dataset specific anomalies.

4.2.2 Generative Modeling

Generative models have been used to generate out of distribution (OOD) data for various applications in semi-supervised learning (Bad GAN [Dai et al., 2017], Margin GAN [Dong and Lin, 2019]), anomaly detection (Fence GAN [Ngo et al., 2019]), OOD detection (BDSG [Dionelis et al., 2020, Du et al., 2022]), medical anomaly detection [Wolleb et al., 2022] and novelty detection [Mirzaei et al., 2023]. However, such methods mostly work with low dimensional data and are not suitable for generating OOD data for VAD. OGNet [Zaheer et al., 2020a, Zaheer et al., 2022a] and G2D [Pourreza et al., 2021] exploit a GAN-based generator and discriminator for VAD. During the first phase of training, a pre-trained state of the generator is used to create PAs while in the second phase, binary classification is performed to distinguish between normal and PAs samples.

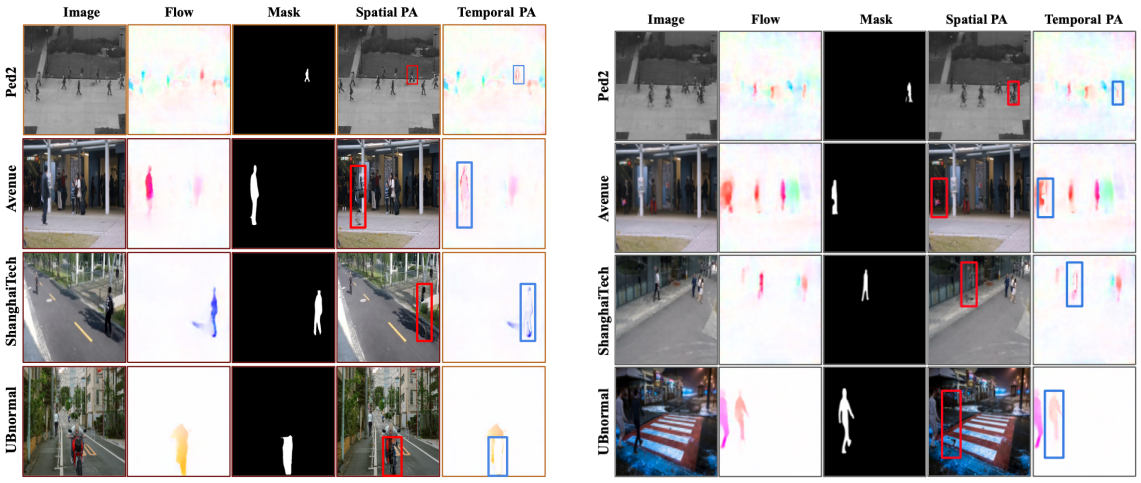
Several VAD works have exploited DMs though their specific methodologies and goals vary. [Tur et al., 2023a, Tur et al., 2023b, Yan et al., 2023] focus on reconstruction and prediction of spatio-temporal and compact motion features extracted from 3D-ResNet/3D-ResNext based encoders using an end-to-end trainable DM. We design our model from the perspective of generating generic spatio-temporal PAs where a generative model (pre-trained LDM) is availed to generate spatial PAs while the mixup method is exploited to create temporal PAs from optical flow.

4.2.3 Other VAD Methods

Non-Reconstruction Based Methods: Various non-reconstruction based methods have also been proposed which derive their anomaly scores from various different indicators of anomaly in addition to reconstruction loss. The work presented in [Liu et al., 2018a] utilised a future frame prediction task for VAD and estimated optical flow and gradient loss as supplementary cues for anomalous behaviour. [Georgescu et al., 2021a, Ionescu et al., 2019a] performed object detection as a pre-processing step under the assumption that anomalous events are always object-centric. Several other works added optical flow components [Ji et al., 2020, Lee et al., 2020b] to detect anomalous motion patterns and a binary classifier [Zaheer et al., 2020a, Pourreza et al., 2021] to estimate anomaly scores. In our work, we also use a segmentation mask and optical flow to generate corresponding spatial and temporal PAs during the training phase. However during inference we do not carry out any object detection and perform anomaly detection solely based on reconstruction of whole images and optical flow.

Non-OCC methods: [Georgescu et al., 2021a] introduced a self-supervised method where different pretext tasks such as arrow of time, middle-box prediction, irregular motion discrimination and knowledge distillation were jointly optimised for VAD. [Wang et al., 2022a] adopted a self-supervised single pre-text task of solving decoupled temporal and spatial jigsaw puzzles Several works have also addressed the VAD problem as a weakly supervised problem through multiple instance learning [Sultani et al., 2018, Wu et al., 2020, Zhu et al., 2022, Zhang et al., 2023a]. Unsupervised VAD methods involve the cooperation of two networks through an iterative process for pseudo-label generation [Zaheer et al., 2022b, Pang et al., 2020, Zaheer et al., 2020c, Zaheer et al., 2020b, Zaheer et al., 2020d, Lin et al., 2022]. Zero-shot VAD was introduced in [Aich et al., 2023] where a model was trained on the source domain to detect anomalies in a target domain without any domain adaptation. USTN-DSC [Yang et al., 2023] a proposed video event restoration framework for VAD while EVAL [Singh et al., 2023] presented a technique for video anomaly

localisation allowing for human interpretable explanations.



(a) (b)

Figure 4.2: Qualitative Assessment: Visualisation of spatial and temporal PAs using segmentation masks. This approach also works with random masks.

4.3 Method

4.3.1 Preliminaries

Latent Diffusion Models (LDMs): Diffusion Probabilistic Models (DMs) [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2020] are a class of probabilistic generative models that are designed for learning a data distribution $p_{\text{data}}(\mathbf{x})$. DMs iteratively denoise a normally distributed variable by learning the reverse process of a fixed Markov Chain of length T through a denoising score matching objective [Song et al., 2020] given by:

$$\mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\tau\sim p_\tau,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[||\mathbf{y}-\mathbf{f}_\theta(\mathbf{x}_\tau;\mathbf{c},\tau)||_2^2], \tag{4.1}$$

where $\mathbf{x}\sim p_{data}$, the diffused input can be constructed by $\mathbf{x}_\tau=\alpha_\tau\mathbf{x}+\sigma_\tau\epsilon$, $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ and is fed into a denoiser model $\mathbf{f}_\theta$, $(\sigma_\tau,\alpha_\tau)$ denotes the noise schedule parameterised by diffusion-time τ, p_τ is a uniform distribution over τ, $\mathbf{c}$ denotes conditioning information and the target vector $\mathbf{y}$ is either the random noise ϵ or $\mathbf{v}=\alpha_\tau\epsilon-\sigma_\tau\mathbf{x}$. The forward diffusion process corresponds to gradual addition of the Gaussian noise

to $\mathbf{x}$ such that the logarithmic signal-to-noise ratio $\lambda_\tau = \log(\alpha_\tau^2/\sigma_\tau^2)$ monotonically decreases.

LDMs [Rombach et al., 2022] were proposed to make standard DMs efficient by training a VQGAN [Esser et al., 2021] based model to project input images i.e. $\mathbf{x} \sim p_{data}$ into a spatially lower dimensional latent space of reduced complexity and then reconstructing the actual input with high accuracy. In particular, a regularised AE [Rombach et al., 2022] is used to reconstruct the input $\mathbf{x}$ such that the reconstruction is given by : $\hat{\mathbf{x}} = \mathbf{f_{de}} \circ \mathbf{f_{en}}(\mathbf{x})$[1] $\approx \mathbf{x}$, where $\mathbf{f_{en}}$ and $\mathbf{f_{de}}$ denotes encoder and decoder respectively. Furthermore an adversarial objective is added using a patch-based discriminator [Isola et al., 2017] to ensure photorealistic reconstruction. DM is then trained in the latent space by replacing $\mathbf{x}$ with its latent representation $\mathbf{z} = \mathbf{f_{en}}(\mathbf{x})$ in eq. (4.1). This leads to reduction in number of learnable parameters and memory.

4.3.2 Generating Spatial-PAs

Real world anomalies are highly context specific without having a ubiquitous definition. Ramachandra *et al.* [Ramachandra et al., 2020b] loosely define them as, the "occurrence of unusual appearance and motion attributes or the occurrence of usual appearance and motion attributes at an unusual locations or times". Examples of such cases include: an abandoned object in a crowded area or suspicious behaviour of an individual. We address this notion of occurrence of unusual appearance attributes through generation of spatial PAs.

Since LDMs achieve state-of-the-art performance on the image inpainting task, they can be exploited as a spatial PAs generator. In particular, we hypothesise that an off-the-shelf pre-trained LDM model [Rombach et al., 2022] without any finetuning on VAD datasets can inpaint the image with enough spatial distortion that can serve as spatially pseudo-anomalous samples for training a VAD model. We follow the mask generation strategy proposed in LAMA [Suvorov et al., 2022] to generate both randomly shaped and object segmentation masks $\mathbf{m}$. We concatenate

[1] $\circ$: denotes function composition

image $\mathbf{x}$, masked image $\mathbf{x} \odot \mathbf{m}$ [2] and mask $\mathbf{m}$ over the channel dimension and give this 7 channel input to UNet [Ronneberger et al., 2015]. We denote the normal data samples as $\mathbf{x}$ unless otherwise explicitly stated. The spatial PAs $\mathcal{P}_s(\mathbf{x})$ is given by:

$$\mathcal{P}_s(\mathbf{x}) = \mathcal{F}_s(\text{stack}(\mathbf{x}, \mathbf{x} \odot \mathbf{m}, \mathbf{m}); \theta), \tag{4.2}$$

where $\mathcal{F}_s$ is the inpainting model that uses latent diffusion with pre-trained model parameters θ. Some examples of the spatial PAs are shown in Figures 4.2a and 4.2b. We avoid regress tuning of LDM hyperparameters due to limited available compute.

4.3.3 Generating Temporal-PAs

We address the notion of unusual motion occurrences (such as person falling to ground) through the generation of temporal PAs. Various video diffusion models [He et al., 2022b, Ho et al., 2022, Voleti et al., 2022] have been proposed, which can be exploited to induce temporal irregularity in the video. However due to limited computational resources, we introduce a simple but effective strategy for the generation of temporal PAs by applying a vicinal risk minimisation technique mixup [Zhang et al., 2018] to the optical flow of the normal videos. More specifically, given a *normal* video $\mathbf{v}$, its frame $\mathbf{x_t}$, and its corresponding segmentation mask $\mathbf{m_t}$ and another consecutive frame $\mathbf{x_{(t+1)}}$, we compute the optical flow $\phi(\mathbf{x_t}, \mathbf{x_{(t+1)}})$ using the TVL1 alogrithm [Zach et al., 2007]. For simplification, we use ϕ as an alias to represent $\phi(\mathbf{x_t}, \mathbf{x_{(t+1)}})$. Let us consider a rectangular patch $\mathbf{p}'$ in ϕ corresponding to the mask $\mathbf{m_t}$ in the frame $\mathbf{x_t}$ with dimensions μ_h and μ_w. In order to perturb the optical flow ϕ, we take another rectangular patch $\mathbf{p_r}'$ at a random location in ϕ with the same dimensions as $\mathbf{p}'$ and apply mixup to yield $\hat{\mathbf{p}}$, which is a convex combination of $\mathbf{p}'$ and $\mathbf{p_r}'$ given by : $\hat{\mathbf{p}} = \lambda \mathbf{p}' + (1 - \lambda)\mathbf{p_r}'$, where λ is sampled from a beta distribution with $\alpha = 0.4$ as in [Zhang et al., 2018]. We denote the temporal PAs as $\mathcal{P}_t(\mathbf{x})$ given by:

[2] $\odot$: denotes point-wise multiplication

$$\mathcal{P}_t(\mathbf{x}) = \mathcal{F}_t(\phi(\mathbf{x_t}, \mathbf{x_{(t+1)}})), \tag{4.3}$$

where $\mathcal{F}_t$ is the temporal PAs generator. Some examples of temporal PAs are depicted in Figure 4.2a. It is important to note that our PAs generation method does not explicitly require segmentation masks, it can also generate PAs using random masks. Since segmentation masks carry semantic meaning, using them enables generation of more semantically informative PAs as further validated by our experiments.

4.3.4 Reconstruction Model

During training regardless of the input ($\mathcal{I}$) i.e normal ($\mathbf{x}/\phi$) or PAs ($\mathcal{P}_s(\mathbf{x})/\mathcal{P}_t(\mathbf{x})$), the network is forced to reconstruct only the normal input using a 3D-CNN (Convolutional Neural Network) based AE model adapted from the convolution-deconvolution network proposed by [Gong et al., 2019a] (Table 4.2).

We train two different AEs with the aim of limiting their reconstruction capacity by exposing them to spatial and temporal PAs. We represent the spatial (temporal) AE by $\mathcal{A}^s(\mathcal{A}^t)$ with $\mathcal{A}^s_{\text{e}}(\mathcal{A}^t_{\text{e}})$ and $\mathcal{A}^s_{\text{de}}(\mathcal{A}^t_{\text{de}})$ denoting its encoder and decoder respectively. The reconstruction output of $\mathcal{A}^s$ is given by : $\hat{\mathbf{x}} = \mathcal{A}^s_{\text{de}} \circ \mathcal{A}^s_{\text{e}}(\mathbf{x})$ while the reconstruction output of $\mathcal{A}^t$ is computed by : $\hat{\phi} = \mathcal{A}^t_{\text{de}} \circ \mathcal{A}^t_{\text{e}}(\phi)$. In order to train $\mathcal{A}^s$ and $\mathcal{A}^t$, PAs ($\mathcal{P}_s(\mathbf{x})$ or $\mathcal{P}_t(\mathbf{x})$) are given as respective inputs with a probability p_s (or p_t) while the normal data is provided as input with probability of $(1 - p_s)$ (or $(1 - p_t)$). p_s (or p_t) is a hyperparameter to control the ratio of PAs to normal samples. Overall, the loss for $\mathcal{A}^s$ and $\mathcal{A}^t$ is calculated as:

$$\mathcal{L}_{\mathcal{A}^{(s)}} = \frac{1}{\Pi} \begin{cases} ||\hat{\mathbf{x}} - \mathbf{x}||_2^2 & \text{if } \mathcal{I} = \mathbf{x} \\ ||\hat{\mathcal{P}}_s(\mathbf{x}) - \mathbf{x}||_2^2 & \text{if } \mathcal{I} = \mathcal{P}_s(\mathbf{x}), \end{cases} \tag{4.4}$$

$$\mathcal{L}_{\mathcal{A}^{(t)}} = \frac{1}{\Pi} \begin{cases} ||\hat{\phi} - \phi||_2^2 & \text{if } \mathcal{I} = \phi \\ ||\hat{\mathcal{P}}_t(\mathbf{x}) - \phi||_2^2 & \text{if } \mathcal{I} = \mathcal{P}_t(\mathbf{x}), \end{cases} \tag{4.5}$$

where $1/\Pi$ is normalisation factor, $\Pi = \mathcal{T} \times C \times H \times W$ and $||.||_2$ is the $\mathcal{L}_2$ norm, where $\mathcal{T}, C, H$ and W are the number of frames, channels, height, and width of frames in the input sequence ($\mathcal{I}$), respectively. The design of $\mathcal{A}^s(\mathcal{A}^t)$ is purposefully chosen to be simplistic (3D-CNN) instead of complex models (vision transformers [Dosovitskiy et al., 2020], 3D ResNets/ResNexts [Hara et al., 2018, Xie et al., 2017]) to explore the degree to which the results can be enhanced by incorporating simple methods.

4.3.5 Estimating Semantic Inconsistency

While measuring the spatial reconstruction quality and temporal irregularity between normal and anomalous data is essential for real-world VAD, it is also crucial to learn and estimate the semantic inconsistency (degree of misalignment of semantic visual patterns and cues) between normal and anomalous samples (e.g. abnormal object in the crowded scene). In practice, to emulate this idea in our approach, we extract frame-level semantically rich features from the ViFi-CLIP [Rasheed et al., 2023] model (pre-trained on Kinetics-400 [Kay et al., 2017]) and perform binary classification between normal data samples $\mathbf{x}$ and spatial pseudo-anomalies $\mathcal{P}_s(\mathbf{x})$ using a discriminator $\mathcal{D}$, (Table 4.1), which can be viewed as an auxiliary component to AEs. Intuitively, it is highly likely that latent space representation of PAs will be semantically inconsistent to the normal scenarios. Our overall architecture is shown in Fig. 4.1.

4.4 Experimental Setup

4.4.1 Implementation Details

(a): UBnormal data-split under OCC Setting. In order to use this dataset in the one class classification (OCC) setting, we train our model using only the normal 186 videos in the training split and the pseudo-anomalies (PAs) generated using them (i.e. totally ignoring the abnormal samples provided in the train set). We tested our model on all the videos in the validation split, comprising of 64 videos with both normal and abnormal events. Such a setting was chosen to keep consistency in evaluation as with other datasets under the OCC setting. The frame-level groundtruth annotation for validation set of UBnormal [Acsintoae et al., 2022] was created using the script[3] provided by the authors.

(b): Pseudo-Anomaly Construction. We take an off-the-shelf Latent Diffusion Model [Rombach et al., 2022] (LDM[4]) pre-trained on the Places dataset [Zhou et al., 2017]. We do not perform any finetuning of the LDM on any video anomaly dataset and therefore it is "under-trained" on video data and hence capable of spatially distorting them. For inpainting the masked out regions of the images, 50 steps of inference were carried out. It is to be noted that due to lack of computational resources we did not experiment with other values of timesteps or any end-to-end finetuning. A very low number of timesteps may produce mostly noisy inpainting output while a very high value might result in inpainted images very close to the input image. The strategy for generation of random and segmentation masks was adopted from the code[5] provided by the authors of LAMA [Suvorov et al., 2022]. If a segmentation mask was not detected for a frame, a random mask was selected instead.

c). Training Spatial ($\mathcal{A}^s$) and Temporal ($\mathcal{A}^t$) AE's: We closely follow the training procedure described in [Astrid et al., 2021a] to train $\mathcal{A}^s$ and $\mathcal{A}^t$. The

[3]https://github.com/lilygeorgescu/UBnormal/tree/main/scripts
[4]https://github.com/CompVis/latent-diffusion/tree/main
[5]https://github.com/advimman/lama/tree/main/saicinpainting

architecture of $\mathcal{A}^s$ and $\mathcal{A}^t$ is adapted from [Gong et al., 2019a], however instead of relying on single channel image as input we use all 3 channels. $\mathcal{A}^s$ and $\mathcal{A}^t$ were trained on respective datasets from scratch with the objective defined in eq. 4.4 and eq. 4.5 respectively on 2 NVIDIA GeForce 2080 Ti GPUs with effective batch size ($\mathcal{B}$) of 24 distributed across the GPUs (12 each). The input to $\mathcal{A}^s$ and $\mathcal{A}^t$ is of size ($\mathcal{B} \times \mathcal{T} \times 3 \times 256 \times 256$), where $\mathcal{T} = 16$. The spatial and temporal PAs were sampled by probability $p_s = 0.4$ and $p_t = 0.5$ respectively. $\mathcal{A}^s$ is trained with Adam optimiser for 25 epochs with a learning rate of 1e$-$4. During training, the reconstruction loss is calculated across all 16 frames of the sequence. The training of the $\mathcal{A}^t$ follows a similar procedure, however the input to the model is the optical flow representing normal events i.e ϕ and temporal PAs $\mathcal{P}_t(\mathbf{x})$.

(d): Extracting ViFi-CLIP Features. For the training split of the benchmark datasets and their corresponding spatial pseudo-anomalies, we extract frame level features using the ViFi-CLIP [Rasheed et al., 2023] model. The input to the ViFi-CLIP model has size : $\mathcal{B}' \times \mathcal{T}' \times 3 \times 224 \times 224$, where $\mathcal{B}'$ (batch size) was set to 1 and $\mathcal{T}'$ (# of frames) was set to 16. All frames were passed into ViFi-CLIP in a sliding window fashion with a stride of 16 therefore we obtain a 512-dimensional feature for every frame. ViFi-CLIP uses the backbone of ViT-B/16 [Dosovitskiy et al., 2020] and is pre-trained on Kinetics-400 [Kay et al., 2017]. It is to be noted that the ViFi-CLIP model performs temporal pooling of the CLIP [Radford et al., 2021] features, however we do not perform temporal pooling and use the frame level representations as during inference we evaluate our pipeline using frame level micro AUC scores. For the frames of the videos in test split (Ped2, Avenue, ShanghaiTech) and validation split (UBnormal), we follow the same procedure for feature extraction.

e). Training the Discriminator ($\mathcal{D}$): During the training phase, the input to $\mathcal{D}$ has a batch size of 16 and feature dimension of 512. The model was trained using a SGD optimiser with a learning rate of 0.02, momentum of 0.9 and weight decay of 10^{-3} for 20 epochs. The groundtruth for normal and PAs samples are given labels 0 and 1 respectively. Figure 4.1 depicts the complete pipeline.

4.4.2 Architectural Details

Table 4.1: Discriminator ($\mathcal{D}$) architecture details

Layers	(Input size, Output size)
Linear Layer 1	(512,128)
ReLU	-
Linear Layer 2	(128,1)

Table 4.2: Autoencoder ($\mathcal{A}^s$ and $\mathcal{A}^t$) architecture details

	Layer	Input Channels	Output Channels	Filter Size	Stride	Padding	Negative Slope
Encoder	Conv3D	3	96	(3,3,3)	(1,2,2)	(1,1,1)	-
	BatchNorm3D	-	-	-	-	-	-
	LeakyReLU	-	-	-	-	-	0.2
	Conv3D	96	128	(3,3,3)	(2,2,2)	(1,1,1)	-
	BatchNorm3D	-	-	-	-	-	-
	LeakyReLU	-	-	-	-	-	0.2
	Conv3D	128	256	(3,3,3)	(2,2,2)	(1,1,1)	-
	BatchNorm3D	-	-	-	-	-	-
	LeakyReLU	-	-	-	-	-	0.2
	Conv3D	256	256	(3,3,3)	(2,2,2)	(1,1,1)	-
	BatchNorm3D	-	-	-	-	-	-
	LeakyReLU	-	-	-	-	-	0.2
Decoder	ConvTranspose3D	256	256	(3,3,3)	(2,2,2)	(1,1,1)	-
	BatchNorm3D	-	-	-	-	-	-
	LeakyReLU	-	-	-	-	-	0.2
	ConvTranspose3D	256	128	(3,3,3)	(2,2,2)	(1,1,1)	-
	BatchNorm3D	-	-	-	-	-	-
	LeakyReLU	-	-	-	-	-	0.2
	ConvTranspose3D	128	96	(3,3,3)	(2,2,2)	(1,1,1)	-
	BatchNorm3D	-	-	-	-	-	-
	LeakyReLU	-	-	-	-	-	0.2
	ConvTranspose3D	96	3	(3,3,3)	(1,2,2)	(1,1,1)	-
	Tanh	-	-	-	-	-	-

4.4.3 Inference

During inference (Figure 4.3), our goal is to *temporally localise the anomaly* by measuring all three types of anomaly indicators of all frames in the test video in the given dataset i.e reconstruction quality, temporal irregularity and semantic inconsistency. Therefore, our anomaly score holistically combines these aspects to gain deeper insights into real-world anomalies in videos.

In order to measure the reconstruction quality, we follow the recent works of [Dong et al., 2020a, Liu et al., 2018a, Park et al., 2020], which utilise normalised Peak Signal to Noise Ratio P_t (PSNR) between the test input frame at time t

and its reconstruction from $\mathcal{A}^s$ to calculate the anomaly score $\omega_1^{(t)}$. The input to $\mathcal{A}^s$ is given in a non-overlapping sliding window fashion with dimensions $1 \times 16 \times 3 \times 256 \times 256$, where batch size is 1 and 16 (window size) represents number of frames. At test time, only the 9^{th} frame of a sequence is considered for anomaly score calculation as in [Astrid et al., 2021a]. For measuring temporal irregularity, a similar strategy is followed as for frames but instead of measuring the PSNR, the normalised $\mathcal{L}_2$ loss (denoted by $\omega_2^{(t)}$) is computed between the input test ϕ at time t and its reconstruction from $\mathcal{A}^t$. For measuring semantic inconsistency, the sequence of input frames is fed into $\mathcal{D}$ in a sliding window fashion (window size = 16). We compute the output probability of a frame at time t to be anomalous from its ViFi-CLIP feature representation and denote it by $\omega_3^{(t)}$. The aggregate anomaly score is given by the weighted average:

$$\omega_{agg}^{(t)} = \begin{cases} \eta_1\omega_1^{(t)} + \eta_2\omega_2^{(t)} + \eta_3\omega_3^{(t)}, & \text{w/ } \mathcal{D} \\ \eta_1\omega_1^{(t)} + \eta_2\omega_2^{(t)}, & \text{w/o } \mathcal{D}; (\eta_3 = 0) \end{cases} \tag{4.6}$$

where η_1, η_2, η_3 are tuned for every dataset. (Refer to section 4.5 for further details). The code for reproducing the results presented in this Chapter is available at `https://github.com/rayush7/unified_PA`.

4.5 Evaluation Criteria

To measure the reconstruction quality, we follow the recent works of [Dong et al., 2020a, Liu et al., 2018a, Park et al., 2020], which utilised normalized Peak Signal to Noise Ratio (PSNR) P_t between an input frame and its reconstruction to calculate the anomaly score. This is illustrated in the following equation.

$$P_t = 10 \log_{10} \frac{M_{\hat{\mathbf{x}}_\mathbf{t}}^2}{\frac{1}{R}||\hat{\mathbf{x}}_\mathbf{t} - \mathbf{x}_\mathbf{t}||_2^2}, \tag{4.7}$$

$$\omega_1^{(t)} = 1 - \frac{P_t - \min_t(P_t)}{\max_t(P_t) - \min_t(P_t)}, \tag{4.8}$$

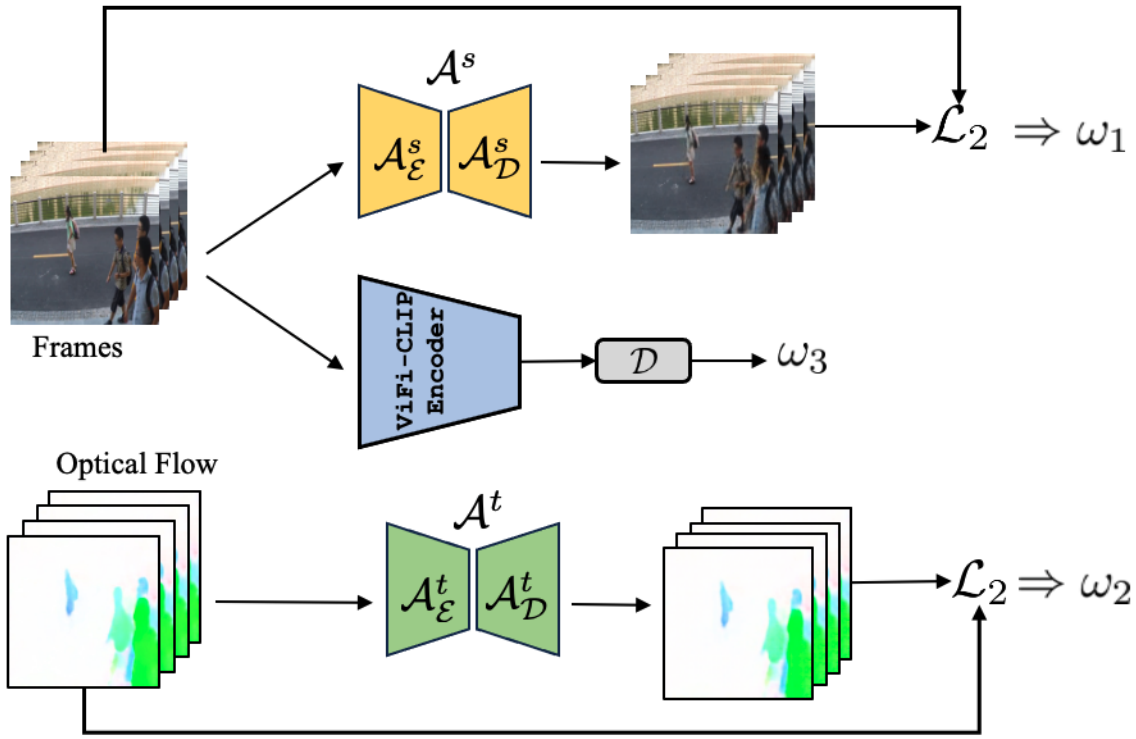


Figure 4.3: During inference, aggregate anomaly score is computed by calculating the weighted sum (eq 4.10) of all the three types of anomaly information; reconstruction quality ω_1 (eq 4.8), temporal irregularity ω_2 (eq 4.9) and semantic inconsistency ω_3.

where $\mathbf{x_t}$ is the input frame at time t, $\hat{\mathbf{x}}_\mathbf{t}$ represents reconstruction of $\mathbf{x_t}$, R denotes the total number of pixels in $\hat{\mathbf{x}}_\mathbf{t}$ and $M_{\hat{\mathbf{x}}_\mathbf{t}}$ is the maximum possible pixel value of $\hat{\mathbf{x}}_\mathbf{t}$. The anomaly score $\omega_1^{(t)}$ is an indicator of reconstruction quality of the input frame. For measuring the temporal irregularity, we compute the normalised $\mathcal{L}_2$ loss between input optical flow at time t and its reconstruction given by the equation:

$$\omega_2^{(t)} = \frac{1}{R'} ||\hat{\phi}(\mathbf{x_t}, \mathbf{x_{(t+1)}}) - \phi(\mathbf{x_t}, \mathbf{x_{(t+1)}})||_2^2, \tag{4.9}$$

where $\phi(\mathbf{x_t}, \mathbf{x_{(t+1)}})$ is the input optical flow frame calculated using consecutive frames $\mathbf{x_t}$ and $\mathbf{x_{(t+1)}}$, $\hat{\phi}(\mathbf{x_t}, \mathbf{x_{(t+1)}})$ represents the reconstruction of $\phi(\mathbf{x_t}, \mathbf{x_{(t+1)}})$, R' denotes the total number of pixels in $\hat{\phi}(\mathbf{x_t}, \mathbf{x_{(t+1)}})$. To measure the semantic inconsistency, the input frames sequence is fed into $\mathcal{D}$ in a sliding window fashion with a window size of 16. The output probability $(\omega_3^{(t)})$ of a frame at time t to be anomalous is computed using its ViFi-CLIP feature representation.

A higher value of $\omega_1^{(t)}$, $\omega_2^{(t)}$ and $\omega_3^{(t)}$ represents higher reconstruction error for

frame and optical flow and high anomaly probability at time t in the test videos during inference. Alternatively, they are indicators of poor reconstruction quality, temporal irregularity and semantic inconsistency and their aggregation can aid in determining real-world anomalies. The aggregate anomaly score is given by the following equation :

$$\omega_{agg}^{(t)} = \begin{cases} \eta_1\omega_1^{(t)} + \eta_2\omega_2^{(t)} + \eta_3\omega_3^{(t)}, & \text{w/ } \mathcal{D} \\ \eta_1\omega_1^{(t)} + \eta_2\omega_2^{(t)}, & \text{w/o } \mathcal{D}; (\eta_3 = 0) \end{cases} \tag{4.10}$$

where η_1, η_2, η_3 are weights assigned to $\omega_1^{(t)}$, $\omega_2^{(t)}$ and $\omega_3^{(t)}$ respectively. The values of η_1, η_2 and η_3 lies in the interval $[0, 1]$ and their sum is equal to 1. We manually tune the values of η_1, η_2, η_3 for all the datasets. The values of (η_1, η_2, η_3) for all the datasets are given by - Ped2 (0.65,0.25,0.1), Avenue (0.45,0.5,0.05), Shanghai (0.85, 0.13, 0.02) and UBnormal (0.4, 0.5, 0.1). In all of the cases, any of the three component can be excluded during evaluation by setting the corresponding weight (η_1, η_2, η_3) to zero.

Note : We also experimented with the learnt weights for the three anomaly indicators but there was a marginal decrease in the performance compared to manually tuning their weights.

Evaluation Metric. For evaluation, we follow the standard metric of frame-level area under the ROC curve (micro-AUC) as in [Zaheer et al., 2020a]. We obtain the ROC curve by varying the anomaly score thresholds to plot False Positive Rate and True Positive Rate for the whole test set for a given dataset. Higher AUC values indicate better performance and more accurate detection of anomalies.

4.5.1 Results

We performed extensive and exhaustive quantitative and qualitative assessments on four datasets namely Ped2 [Li et al., 2014], Avenue [Lu et al., 2013], ShanghaiTech [Luo et al., 2017c] and UBnormal [Acsintoae et al., 2022].

Baselines: We compare our results with memory based AE [Gong et al., 2019b,

Table 4.3: Micro AUC score comparison between our approach and existing state-of-the-art methods on val split of UBnormal [Acsintoae et al., 2022].

Reconstruction Methods	UBnormal [Acsintoae et al., 2022]
Baseline (without PAs)	54.06 %
LNTRA Astrid *et al.* [Astrid et al., 2021a] - Patch based	57.09 %
LNTRA Astrid *et al.* [Astrid et al., 2021a] - Skip-frame based	55.48 %
Ours w/o $\mathcal{D}$	57.53 %
Ours w/ $\mathcal{D}$	**57.98 %**

Park et al., 2020] and other reconstruction based method trained with pseudo-anomalous samples created using other simulation techniques [Astrid et al., 2021a, Astrid et al., 2021b]. The network trained without any PAs is represented as the standard *baseline*. The model design of the AE is fixed across all the experimental settings. Object-level information is only considered for perturbing the normal data during training while at inference we evaluate results strictly based on reconstruction and classification outputs i.e. without any object detection. Hence our method is not directly comparable to object-centric methods.

1. Quantitative Assessment: In Table 4.4, we report micro AUC comparisons of overall scores of our model and existing SOTA methods on test sets of Ped2, Avenue and Shanghai datasets. We follow the same practice as in [Astrid et al., 2021a] of dividing the SOTA methods into 5 categories - 1) Non Deep Learning 2) Object centric approaches which formulates VAD based on anomalous behaviour of objects and employs the use of an object detector during training and inference 3) Prediction based method that performs the task of next frame prediction 4) Reconstruction based techniques follow the strategy of reconstructing the inputs 5) Miscellaneous, which do not lie within any of these categories.

Our method is closest to reconstruction based methods though we also avail the discriminator $\mathcal{D}$ as the auxiliary component to learn the distance between normal data distribution and PAs distribution. For clarity, we provide results *with and without* $\mathcal{D}$ for all the datasets. Compared to memory-based networks, our unified framework trained on synthetically generated spatio-temporal PAs outperforms MemAE [Gong et al., 2019b] and MNAD-Reconstruction [Park et al., 2020] on Avenue and Shanghai while on Ped2 surpasses MNAD-Reconstruction and achieves comparable performance as MemAE. We also compare our results with

other PAs generator methods such as STEAL Net [Astrid et al., 2021b] and LNTRA [Astrid et al., 2021a]. We observe that on the Avenue dataset our model outperforms LNTRA (patch, skip-frame based) though marginally lags behind STEAL-Net whereas STEAL-Net and LNTRA achieve better performance than our model on Ped2 and Shanghai dataset. However such methods generate PAs under bold assumptions and inductive biases which may cause them to fail in particular cases. We report such cases in the Ablation study (Figure 4.6). In Table 4.5 we show that the transfer performance of our model is comparable with other PAs generation methods (see section 4.5.2). We do employ optical flow like other methods (Frame-Pred [Liu et al., 2018a]) and observe that our results outperform Frame-Pred on the Avenue, achieve comparable performance on ShanghaiTech and are marginally less on Ped2.

In Table 4.3, we show a comparison between baseline, [Astrid et al., 2021a] and our approach on the validation set of the UBnormal dataset by training only on the normal videos in the train split. This is done to ensure consistency in evaluation under the OCC setting. The training and evaluation for baseline and LNTRA (patch, skip-frame) based methods on UBnormal was performed using scripts provided by the authors of LNTRA[6]. We observe that our method outperforms baseline and LNTRA achieving micro AUC score of 57.98% and implying that our PAs are generic and applicable for more diverse anomalous scenarios. Both in Table 4.4, 4.3 we notice that the effect of adding $\mathcal{D}$ is minimal, which validates the intuition that VAD cannot be directly addressed as a classification problem.

Table 4.4, 4.3 show that no single reconstruction-based method excels on all datasets. This is because anomalies are context-dependent. Different methods have inductive biases that work for specific datasets but not others. Our work provides a generic solution towards generating PAs without making bold assumptions about dataset's anomalies.

2. Qualitative Assessment: We conduct qualitative analysis of the anomaly score over time for sample videos in Avenue, Shanghai (Figure 4.4) and Ped2, UB-

[6]https://github.com/aseuteurideu/LearningNotToReconstructAnomalies

Table 4.4: Micro AUC score comparison between our approach and state-of-the-art methods on test split of Ped2 [Li et al., 2014], Avenue (Ave) [Lu et al., 2013] and ShanghaiTech (Sh) [Luo et al., 2017c]. Best and second best performances are highlighted as **bold** and underlined, in each category and dataset.

	Methods	Ped2 [Li et al., 2014]	Ave [Lu et al., 2013]	Sh [Luo et al., 2017c]
Non deep learning	MDT [Mahadevan et al., 2010]	82.90%	-	-
	Lu *et al.* [Lu et al., 2013]	-	**80.90%**	-
	AMDN [Xu et al., 2017]	90.80%	-	-
	Del Giorno *et al.* [Del Giorno et al., 2016]	-	78.30%	-
	LSHF [Zhang et al., 2016]	**91.00%**	-	-
	Xu *et al.* [Xu et al., 2014]	88.20%	-	-
	Ramachandra and Jones [Ramachandra and Jones, 2020]	88.30%	72.00%	-
Miscellaneous	OLED [Jewell et al., 2022]	99.02%	-	-
	AbnormalGAN [Ravanbakhsh et al., 2017]	93.50%	-	-
	Smeureanu *et al.* [Smeureanu et al., 2017]	-	84.60%	-
	AMDN [Xu et al., 2015, Xu et al., 2017]	90.80%	-	-
	STAN [Lee et al., 2018b]	96.50%	87.20%	-
	MC2ST [Liu et al., 2018b]	87.50%	84.40%	-
	Ionescu *et al.* [Ionescu et al., 2019b]	-	88.90%	-
	BMAN [Lee et al., 2019b]	96.60%	**90.00%**	76.20%
	AMC [Nguyen and Meunier, 2019]	96.20%	86.90%	-
	Vu *et al.* [Vu et al., 2019]	**99.21%**	71.54%	-
	DeepOC [Wu et al., 2019]	-	86.60%	-
	TAM-Net [Ji et al., 2020]	98.10%	78.30%	-
	LSA [Abati et al., 2019]	95.40%	-	72.50%
	Ramachandra *et al.* [Ramachandra et al., 2020a]	94.00%	87.20%	-
	Tang *et al.* [Tang et al., 2020a]	96.30%	85.10%	73.00%
	Wang *et al.* [Wang et al., 2020b]	-	87.00%	**79.30%**
	OGNet [Zaheer et al., 2020a]	98.10%	-	-
	Conv-VRNN [Lu et al., 2019]	96.06%	85.78%	-
	Chang *et al.* [Chang et al., 2020]	96.50%	86.00%	73.30%
	USTN-DSC [Yang et al., 2023]	98.10%	89.90%	73.8%
	EVAL [Singh et al., 2023]	-	86.06%	76.63%
Object-centric	MT-FRCN [Hinami et al., 2017]	92.20%	-	-
	Ionescu *et al.* [Ionescu et al., 2019a] [7]	94.30%	87.40%	78.70%
	Doshi and Yilmaz [Doshi and Yilmaz, 2020a, Doshi and Yilmaz, 2020b]	97.80%	86.40%	71.62%
	Sun *et al.* [Sun et al., 2020]	-	89.60%	74.70%
	VEC [Yu et al., 2020]	97.30%	90.20%	74.80%
	Georgescu *et al.* [Georgescu et al., 2021b]	**98.70%**	**92.30%**	**82.70%**
Prediction	Frame-Pred [Liu et al., 2018a]	95.40%	85.10%	72.80%
	Dong *et al.* [Dong et al., 2020b]	95.60%	84.90%	73.70%
	Lu *et al.* [Lu et al., 2020]	96.20%	85.80%	**77.90%**
	MNAD-Pred [Park et al., 2020]	97.00%	88.50%	70.50%
	AnoPCN [Ye et al., 2019a]	96.80%	86.20%	73.60%
	AMMC-Net [Cai et al., 2021]	96.90%	86.60%	73.70%
	DLAN-AC [Yang et al., 2022b]	**97.60%**	**89.90%**	74.70%
Reconstruction	AE-Conv2D [Hasan et al., 2016]	90.00%	70.20%	60.85%
	AE-Conv3D [Zhao et al., 2017b]	91.20%	71.10%	-
	AE-ConvLSTM [Luo et al., 2017b]	88.10%	77.00%	-
	TSC [Luo et al., 2017c]	91.03%	80.56%	67.94%
	StackRNN [Luo et al., 2017c]	92.21%	81.71%	68.00%
	MemAE [Gong et al., 2019b]	94.10%	83.30%	71.20%
	MNAD-Recon [Park et al., 2020]	90.20%	82.80%	69.80%
	Baseline (without PAs)	92.49%	81.47%	71.28%
	STEAL Net [Astrid et al., 2021b]	**98.40%**	**87.10%**	73.70%
	LNTRA Astrid *et al.* [Astrid et al., 2021a] - Patch based	94.77%	84.91%	72.46%
	LNTRA Astrid *et al.* [Astrid et al., 2021a] - Skip-frame based	96.50%	84.67%	**75.97%**
	Ours w/o $\mathcal{D}$	93.52%	86.51%	71.76%
	Ours w/ $\mathcal{D}$	93.53%	86.61%	71.65%

normal (Figure 4.5). We also compare our model's anomaly score over time with those obtained from LNTRA (skip-frame, patch-based). It can be concluded that on the Avenue and Ped2 datasets, our method detects anomalies fairly well and performance is equivalent with LNTRA models. Though there exist certain failure cases in the Shanghai and UBnormal datasets which occur due to anomalies occurring due to abnormal interaction between two objects i.e. fighting between two individuals in Shanghai and accident with a bike in UBnormal. Though our PAs generator is

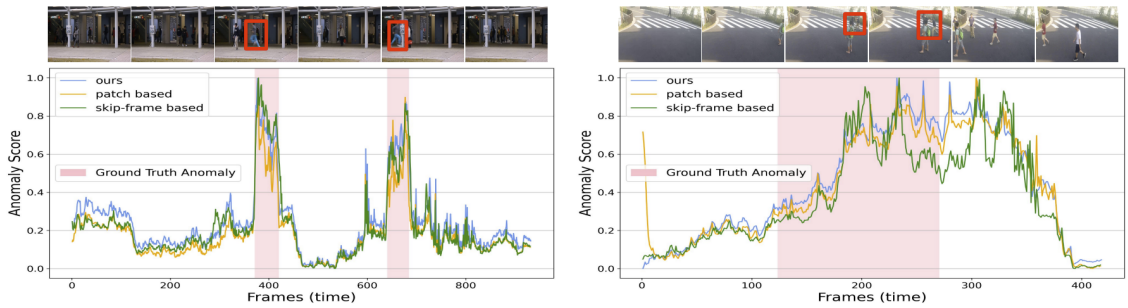


Figure 4.4: Qualitative Assessment : Visualisation of anomaly score over time for sample videos in Avenue (left) and ShanghaiTech (right), compared with other PAs generator and reconstruction based methods in LNTRA [Astrid et al., 2021a] - patch and skip-frame based.

generic, end-to-end finetuning is further needed to emulate such complex real-world anomalies.

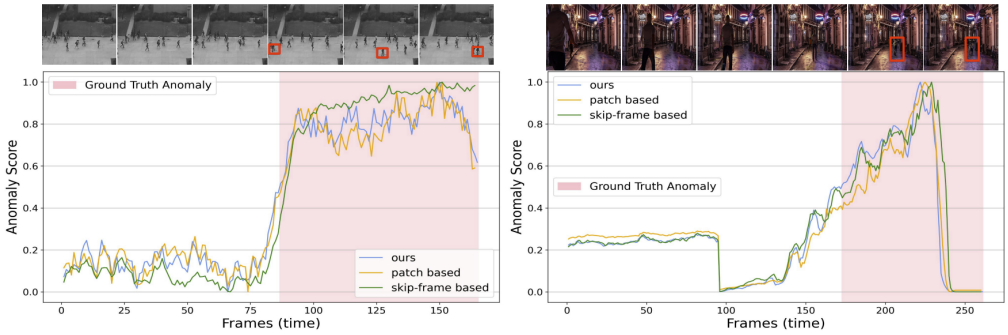


Figure 4.5: Qualitative Assessment : Visualization of anomaly score over time for sample videos in Ped2 (left) and UBnormal (right), compared with other PAs generator and reconstruction based methods in LNTRA [Astrid et al., 2021a] - patch and skip-frame based.

4.5.2 Ablation Studies

1: How transferable are PAs? We also examine how well PAs transfer across various VAD datasets. We use our pre-trained model on UBnormal dataset, which contains a wide range of anomalies and backgrounds, making it suitable for transferability. We tested the model on rest of the datasets without fine-tuning. Our results in Table 4.5 show that our model outperforms the patch-based method on all other datasets while achieves competitive performance compared to the skip-frame based method. This provides an interesting insight that our PAs are generic

and transferable.

Table 4.5: Transfer Performance : micro-AUC scores.

Method	Ped2	Avenue	Shanghai
Patch [Astrid et al., 2021a]	78.80 %	43.94 %	61.57 %
Skip-Frame [Astrid et al., 2021a]	85.21 %	83.82 %	70.52 %
Ours w/ $\mathcal{D}$	85.37 %	83.50 %	70.07 %

2: How to interpret PAs? In Figure 4.6, we compare error heatmaps generated using a model trained with patch and skip-frame based PAs and with our spatial-PAs on all the respective datasets. Since skip-frame and patch based PAs carry strong assumptions, they tend to have problems detecting complicated real-world anomalies in ShanghaiTech such as a baby carriage (anomalous object) whereas our model trained with spatial-PAs yields high error for such cases. Furthermore, our PAs also give strong results on the synthetic dataset UBnormal, where patch and skip-frame based PAs fail to detect complex violent scenes as temporal irregularity induced through skip-frames is not generic. However, even our spatial-PAs, which are not explicitly trained to detect temporal anomalies are able to determine such real-world anomalies. On Avenue and Ped2 datasets, our model yields comparable error to patch based PAs for an anomalous activity however we observe that skip-frame based PAs overly estimates the reconstruction error for the same. Intuitively this indicates that even though skip-frame performs reasonably well on benchmark datasets but it is susceptible to amplification of the error. An explanation for this phenomena could be due to underlying strong assumption of skipping frames based on a specific stride value to model temporal irregularity. These observations validate that our PAs are generalised and enable understanding of which real-world anomalies can be detected using which type of PAs.

3: Random vs Segmentation masks: Table 4.6 shows the effect of using random and segmentation masks for generating spatial PAs. We observe that using a segmentation mask gives better AUC score on Ped2 and Avenue dataset, which is intuitively justified as segmentation masks contain more semantic information. Despite this, our method is flexible in terms of type of mask chosen.

4: Effect of changing the probability of sampling PAs. We conduct an

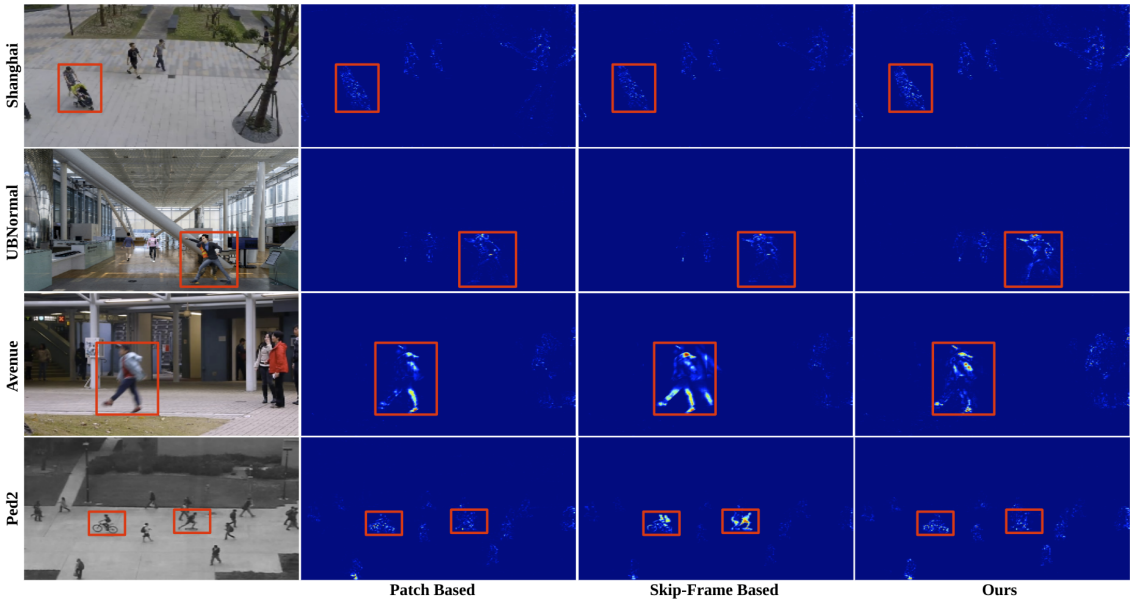


Figure 4.6: Visualisation of error heatmap for sample videos compared with other PAs generator methods in LNTRA [Astrid et al., 2021a].

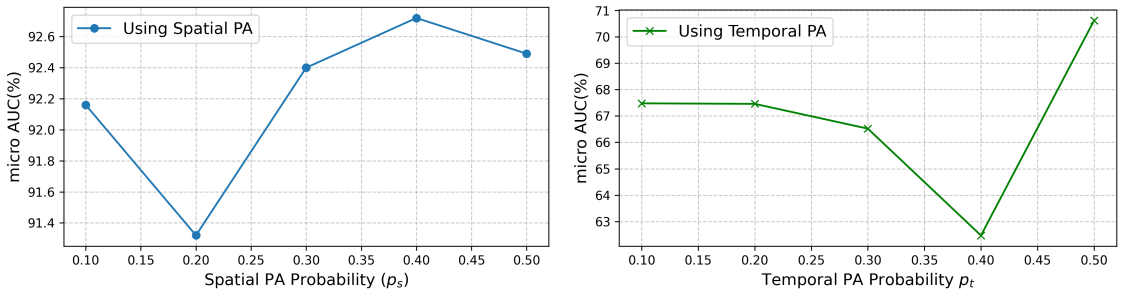
Table 4.6: Effect of Random and Segmentation masks on micro-AUC scores, using the output of $\mathcal{A}^s$ when trained with $p_s = 0.4$.

Mask Type	Ped2	Avenue
Random Mask	91.18 %	83.13 %
Segmentation Mask	92.71 %	84.51 %

experimental study by varying the probability of sampling spatial and temporal PAs (p_s, p_t) on Ped2 during training between 0.1 to 0.5 and measuring micro AUC scores during inference. Figure 4.7 shows that the model achieves best performance when $p_s = 0.4$ and $p_t = 0.5$.

4.6 Conclusions

This Chapter addresses hypothesis $\mathbf{H_2}$ (Chapter 1), which deals with the idea of instilling robustness to spatio-temporal perturbations into the learned video representation by exposing the DL models to near-distribution samples (referred as PAs in our study). We examine the robustness attribute of video representation by exploring the problem of VAD (Chapter 2,4) under the OCC setting. In particular, the aim of this chapter is to answer $\mathbf{R_3}$: *Is it possible to synthetically generate generic PAs by introducing spatio-temporal distortions into normal data in order to*



(a) (b)

Figure 4.7: Comparison of micro-AUC scores on Ped2 dataset calculated from output of $\mathcal{A}^s$ ($\mathcal{A}^t$) trained on a range of values of p_s (p_t) between $\{0.1,0.5\}$. We observe that setting $p_s = 0.4$ and $p_t = 0.5$ yields the best performance as shown in (a) and (b) respectively. These probability values are fixed for all other experiments.

detect real-world anomalies effectively? Furthermore, can such PAs transfer across multiple VAD datasets? and $\mathbf{R_4}$: *How can we design a VAD pipeline that aggregates different anomaly indicators to create a unified anomaly scoring mechanism that effectively captures spatial, temporal, and semantic inconsistencies?.*

To summarise, in this Chapter we presented a novel and generic spatio-temporal PAs generator vital for VAD tasks without incorporating strong inductive biases. We achieve this by adding perturbation in the frames of normal videos by inpainting a masked out region using a pre-trained LDM and by distorting optical flow by applying mixup-like augmentation (Figure 4.2a,4.2b) (**answering $\mathbf{R_3}$**). We also introduced a simple unified VAD framework that learns three types of anomaly indicators i.e. reconstruction quality, temporal irregularity and semantic inconsistency in an OCC setting (Figure 4.1) (**answering $\mathbf{R_4}$**). Extensive evaluation shows that our framework is not objectively SOTA but achieves comparable performance to other SOTA reconstruction methods and PA generators with predefined assumptions across multiple datasets (Table 4.4, 4.3) without any end-to-end finetuning or any post-processing. This indicates the effectiveness, generalisation and transferability of our PAs.

However, there are limitations with the work presented in this Chapter. First, due to limited computational resources our model was not trained end-to-end and doesn't avail more powerful architectures (vision transformers or 3D-ResNets), which

might boost the performance. It would also be interesting to make this setting adaptive by learning a policy network to select which anomaly indicator among poor reconstruction quality, temporal irregularity and semantic inconsistency contributes more towards detection of real-world anomalies. Second, the notion of generating latent space PAs through LDMs or manifold mixup remains to be investigated.

Chapter 5

Efficient Video Representation Learning

This Chapter focuses on *efficient* video representation learning. In this Chapter we inspect $\mathbf{H_3}$ (R_5, R_6), as introduced in Chapter 1, which presumes that incorporating adaptive computation strategies into the self-supervised training objective can promote the learning of transferable and generalizable video representations in a more efficient manner compared to those learned with static computation. To evaluate $\mathbf{H_3}$, we design a Trajectory-Aware Adaptive Token Sampler module that dynamically learns to select the most relevant motion-centric space-time tokens for the self-supervised pre-training objective of MVM. The quality of the learned representation is evaluated using the downstream task of action recognition on benchmark datasets.

5.1 Motivation

Masked video modeling (MVM) has emerged as a highly effective pre-training strategy for visual foundation models, whereby the model reconstructs masked spatiotemporal tokens using information from visible tokens. However, a key challenge in such approaches lies in selecting an appropriate masking strategy. Previous studies have explored predefined masking techniques, including random and tube-based mask-

ing, as well as approaches that leverage key motion priors, optical flow and semantic cues from externally pre-trained models. In this work, we introduce a novel and generalizable **T**rajectory-Aware **A**daptive **T**oken **S**ampler (TATS), which models the motion dynamics of tokens and can be seamlessly integrated into the masked autoencoder (MAE) framework to select motion-centric tokens in videos. Additionally, we propose a unified training strategy that enables joint optimization of both MAE and TATS from scratch using Proximal Policy Optimization (PPO). We show that our model allows for aggressive masking without compromising performance on the downstream task of action recognition while also ensuring that the pre-training remains memory efficient. Extensive experiments of the proposed approach across four benchmarks, including Something-Something v2, Kinetics-400, UCF101, and HMDB51, demonstrate the effectiveness, transferability, generalization, and efficiency of our work compared to other state-of-the-art methods.

Self-supervised video representation learning has recently emerged as a prominent area of research due to the generalization capabilities of the learned embeddings. Such representations can be applied to several downstream tasks such as action recognition [Wang et al., 2022b, Han et al., 2020b], object detection [Akiva et al., 2023], and segmentation [Aydemir et al., 2023] in videos. Due to the scarcity of labeled data, a standard approach in self-supervised learning (SSL) methods for video understanding involves defining a pretext task. A pretext task can be interpreted as a self-supervised pseudo-objective for pre-training a model. Intuitively, if a model learns to solve a complex task that requires a high-level understanding of its input, then the features learned as a result should generalize well to other tasks.

Inspired by BERT [Kenton and Toutanova, 2019] used in language modeling, masked modeling in the form of masked autoencoders (MAE) has been adopted for images [Wei et al., 2022, He et al., 2022a, Li et al., 2022c] and for videos [Feichtenhofer et al., 2022, Tong et al., 2022, Wang et al., 2023a] as a self-supervised pretext task. Masked modeling involves masking a large fraction (between 75-95%) of the input data and then learning to reconstruct or predict the removed content based

on the visible information. Although this concept is simple, it has been shown to improve performance [Feichtenhofer et al., 2022, He et al., 2022a], generalization [Feichtenhofer et al., 2022, He et al., 2022a], data efficiency [Tong et al., 2022], memory efficiency [Feichtenhofer et al., 2022, Bandara et al., 2023], scalability [Wang et al., 2023a, He et al., 2022a], robustness [Hendrycks et al., 2019] and to reduce overfitting [Girdhar et al., 2023] on downstream tasks.

Several studies have explored different formulations of MAE, focusing on masking portions of input, features, or augmenting the masked modeling objective [Bao et al., 2022, Dong et al., 2023, Li et al., 2022c, Xie et al., 2022, Zhou et al., 2022, Xie et al., 2023, Wei et al., 2022]. However, less emphasis has been given to adaptive masking mechanisms that adaptively select space-time patches based on the input. The masking mechanism forms a crucial component of the family of MAE methods, as it is responsible for selecting which information is to be exploited by the encoder and predicted by the decoder.

[He et al., 2022a, Xie et al., 2022] explored random masking approaches for image patches, blocks, and grids. Though such approaches have shown promise and performance gains on downstream tasks, there still exists a research gap in terms of the masking mechanism adapting to the input. With fixed masking mechanisms, MAEs are unable to exploit the expressivity of transformer-based encoders. In this direction, other contemporary works have investigated different masking strategies for images such as semantically guided masking [Li et al., 2022c], uniform sampling for pyramid-based vision transformer (ViT) [Li et al., 2022d] and utilizing mask generators based on object priors [Chen et al., 2023] and learning easy-to-hard masking through curriculum learning [Madan et al., 2024].

The challenging aspect of MVM is the extra time dimension and high spatiotemporal inductive biases from adjacent frames carrying highly redundant information. This introduces potential information leakage as masked space-time patches can be trivially inferred from spatiotemporal neighborhoods, enabling learning of shortcuts and less generalizable representations during pre-training. Hence, a substantial

amount of compute and memory resources are inefficiently utilized in the prediction of uninformative tokens. On the contrary, such a high level of redundancy can be exploited to aggressively mask space-time tokens with a high mask ratio without compromising the prediction quality of the masked space-time patches.

Several approaches in MVM have utilized frame, tube, and patch-based masking [Feichtenhofer et al., 2022, Tong et al., 2022, Wang et al., 2023a], and there is no single universal masking strategy that works for all datasets. VideoMAE [Tong et al., 2022] achieves the best action classification results on Something-Something v2 (SSv2) [Goyal et al., 2017] with random tube masking while STMAE [Feichtenhofer et al., 2022] achieves its best performance on Kinetics-400 (K400) [Kay et al., 2017] with random space-time patch masking. An explanation for this observation is that not all space-time tokens carry meaningful information, and a fixed masking strategy steers the model's optimization towards a particular task. Thus, it is crucial to incorporate adaptive computation in MAEs to dynamically select informative tokens based on the given input and the mask ratio. Previous works such as MGMAE [Huang et al., 2023] proposed motion-guided masking by extracting the optical flow from pre-trained models, and AdaMAE [Bandara et al., 2023] introduced a token sampler module to select high-activity regions using REINFORCE [Williams, 1992].

In order to exploit unequal information density among patches, we introduce the *TATS* module that learns a video-specific masking strategy from scratch to select space-time patches based on their spatio-temporal motion and trajectory information using Trajectory Attention (TA) [Patrick et al., 2021]. *TATS* does not rely on any computationally expensive dense optical flow features or semantic cues obtained from external pretrained models like RAFT [Teed and Deng, 2020], CLIP [Radford et al., 2021], or DINOv2 [Oquab et al., 2023].

TATS can be interpreted as a policy agent that models a categorical distribution over the set of input space-time tokens by leveraging their trajectory information and then sampling the most relevant tokens based on a predefined mask ratio. However, since training MAE in conjunction with *TATS* is unstable due to the non-

differentiability of the sampling operation, we additionally propose a unified training recipe to train MAE and TATS modules simultaneously using the PPO [Schulman et al., 2017] method used in reinforcement learning (RL). Our goal is to incorporate adaptivity into MAEs while preserving their representation quality in terms of generalization and ensuring that the pre-training process remains memory efficient.

Overall, our main contributions in this chapter are:

- We propose a novel and generalizable TATS module that learns to adaptively sample motion-centric tokens for MAE pre-training by modeling their motion trajectories in videos. TATS can be seamlessly integrated into the MAE framework and does not rely on auxiliary modalities like optical flow (RAFT [Teed and Deng, 2020]) or external pre-trained models (DINOv2 [Oquab et al., 2023], CLIP [Radford et al., 2021]) for motion priors or semantic cues.

- Additionally, we introduce a unified training recipe (Algorithm 1) that facilitates the joint optimization of both MAE and TATS from scratch using PPO [Schulman et al., 2017] to ensure stable convergence during pre-training even with aggressive masking.

- Finally, we conduct a comprehensive evaluation on four benchmark datasets (K400, SSv2, UCF101, HMDB51) for action recognition to demonstrate the effectiveness, generalization, transferability, and efficiency of our work compared to the state-of-the-art methods (Tables 5.1,5.2,5.9).

5.2 Related Work

5.2.1 SSL for video representation learning.

SSL has emerged as a promising alternative to the supervised paradigm for pre-training deep models, enabling training on large-scale datasets with enhanced generalization while eliminating the need for labeled annotations. SSL in video primarily focuses on leveraging the temporal dimension for designing tasks such as tempo-

ral ordering [Fernando et al., 2017, Lee et al., 2017, Misra et al., 2016, Wei et al., 2018, Wang et al., 2019a], future prediction [Vondrick et al., 2016, Mathieu et al., 2016, Lotter et al., 2017, Vondrick et al., 2018, Diba et al., 2019], spatiotemporal contrast [Feichtenhofer et al., 2021, Han et al., 2019, Qian et al., 2021b, Sun et al., 2019], temporal coherence [Goroshin et al., 2015, Wiskott and Sejnowski, 2002] and object motion [Agrawal et al., 2015, Pathak et al., 2017, Wang and Gupta, 2015, Wang et al., 2019c].

5.2.2 Masked Modeling.

Masked Language Modeling has been universally adopted in natural language understanding, leading to groundbreaking works such as BERT [Kenton and Toutanova, 2019]. Several researchers have adopted masked prediction for images/videos through Masked Image Modeling (MIM)/MVM, respectively. MIM/MVM can be interpreted as a generalized Denoising Autoencoder [Vincent et al., 2008b] where the masking can be attributed to noise addition.

Generative Pre-training from pixels [Chen et al., 2020a] introduced the task of masked pixel prediction. However, pixel-level prediction demands high computational costs for pre-training and results in inferior performance compared to ConvNets. The notion of dividing an image into visual tokens through patches, as introduced in the ViT [Dosovitskiy et al., 2020], enabled the adoption of BERT-style pre-training for visual tokens. BeiT [Bao et al., 2022] and PeCo[Dong et al., 2023] are built upon using an offline tokenizer to learn discrete codebooks using VQ-VAE [Van Den Oord et al., 2017] with the goal of reconstructing the original image from randomly masked discrete tokens. iBOT [Zhou et al., 2022] and DALL-E [Ramesh et al., 2021] proposed an online tokenizer based on teacher networks trained via self-distillation. Maskfeat [Wei et al., 2022] introduced reconstruction of Histogram-of-Oriented-Gradients features for masked-out regions. MAE [He et al., 2022a] and SimMIM [Xie et al., 2022] claimed that directly reconstructing the RGB pixel values performs equivalent to codebook-based methods. [Zhang et al., 2022] proposed a

theoretical understanding of the masking mechanism.

MVM techniques have been employed for video pre-training by masking random space-time patches as in STMAE [Feichtenhofer et al., 2022], or by utilizing tube masking with a high masking ratio, as in VideoMAE [Tong et al., 2022, Wang et al., 2023a]. Other MVM approaches include BEVT [Wang et al., 2022b], Masked Video Distillation [Wang et al., 2023b]. Our method is specifically designed for videos but can be integrated into the MAE framework while maintaining the original reconstruction target and MAE architecture without any modifications.

5.2.3 Masking Strategies in MIM/MVM.

Many studies have demonstrated that the performance of MAEs and their variants on downstream tasks relies heavily on the choice of masking strategy. In fact, the masking strategy is one of the core design choices in MIM and MVM, and it significantly governs the information that the network learns during pre-training. SemMAE [Li et al., 2022c] harnesses iBOT [Zhou et al., 2022] for semantic segmentation and generates semantically aware masks for pre-training. ADIOS [Shi et al., 2022] introduces a method to jointly learn a masking function and an image encoder through an adversarial objective. AutoMAE [Chen et al., 2023] avails an adversarially-trained mask generator based on Gumbel-softmax [Jang et al., 2016] for MIM. CL-MAE [Madan et al., 2024] uses curriculum learning to generate adaptive masks based on the desired level of complexity (i.e. easy to hard masks). Cluster Masking [Wei et al., 2024] learns to apply random masking to clusters of image patches, while R-MAE [Nguyen et al., 2024] focuses on masking pixels within a specified region. [Li et al., 2022d] proposed a uniform masking strategy that enables MAE pre-training for Pyramid-based ViTs with locality. AttnMask [Kakogeorgiou et al., 2022] proposed a distillation-based MIM where masking of the student network is guided by attention maps generated by the teacher network. [Xie et al., 2023] introduced a method to mask the frequency domain representation of the images using low/high pass filters while [Feng and Zhang, 2023] constructs a patch associa-

tion graph using attention maps and addresses the unlabeled part partition problem as a graph cut problem using the Expectation-Maximization algorithm [Banerjee et al., 2005] to obtain semantics-based masks.

The masking strategy is a core design choice in MVM, which significantly impacts the information that the network learns during pre-training. MGMAE [Huang et al., 2023] and MGM [Fan et al., 2023] introduced motion-guided masking by exploiting a pre-trained lightweight optical flow estimator RAFT [Teed and Deng, 2020] and motion vectors stored in the H.264 codec to select space-time patches with rich motion information. EVEREST [Hwang et al., 2024] proposed redundancy robust token selection and an information-intensive frame selection mechanism for pre-training and fine-tuning. MME [Sun et al., 2023] modifies the pre-training objective from the reconstruction of the appearance content to the reconstruction of the motion trajectory. AdaMAE [Bandara et al., 2023], the work most closely related to ours, proposed an end-to-end trainable token sampling module that learns to sample space-time patches from high-activity regions using REINFORCE [Williams, 1992]. Our approach draws inspiration from AdaMAE [Bandara et al., 2023], however our *TATS* module selects space-time tokens based on their motion trajectories in videos. Additionally, we propose a novel training recipe that jointly optimizes MAE and *TATS* from scratch using PPO, ensuring stable convergence during pre-training, even with aggressive masking.

5.3 Method

5.3.1 Overview of MVM

Here we briefly describe important components of a standard MVM method.

Tokenizer. Consider an input video V of size $T \times C \times H \times W$, where T represents the number of frames, C denotes the input channels and H, W is the height and width of a frame. A *Tokenizer* composed of 3D convolutional layer with kernel K of size (t, C, h, w), stride S of size (t, h, w) and d output channels is availed to tokenize

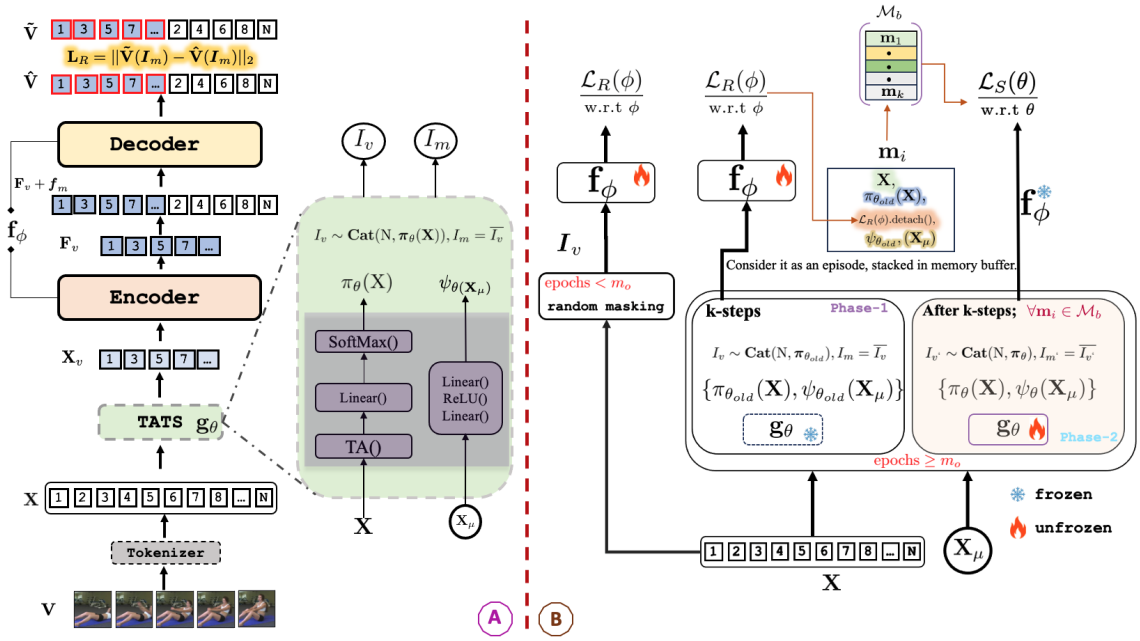


Figure 5.1: A depicts our overall architecture with MAE (f_ϕ) and TATS (g_θ). B illustrates the joint training (Algorithm 1) of f_ϕ and g_θ using PPO. Until epoch m_o, standard random space-time masking is applied. Afterward, every k steps, Phase 1 (g_θ frozen,f_ϕ unfrozen) stores old state of g_θ in memory buffer $\mathcal{M}_b$ as episodes, followed by Phase 2 (g_θ unfrozen,f_ϕ frozen), where g_θ is optimized via $\mathcal{L}_s(\theta)$. The optimization process then alternates between Phase 1 and Phase 2.

V into N number of tokens with dimension d indicated as $\mathbf{X}$, where $N = \frac{T}{t} \times \frac{H}{h} \times \frac{W}{w}$. Positional information is further embedded into the tokens using a fixed 3D periodic positional encoding scheme outlined in [Vaswani et al., 2017].

Token Sampler. Based on a specific masking mechanism (tube [Tong et al., 2022], adaptive [Bandara et al., 2023], random space-time [Feichtenhofer et al., 2022]), a set of visible token indices $\boldsymbol{I}_v$ are sampled from $\boldsymbol{X}$ for a given mask ratio $\rho \in (0, 1)$ while the remaining indices correspond to the masked Indices $\boldsymbol{I}_m$. The choice of masking mechanism is a pivotal design choice of MVM techniques.

Encoder-Decoder. The design of encoder-decoder is usually a variant of VideoMAE [Tong et al., 2022]. The encoded representation $\mathbf{F}_v$ is learned by feeding the sampled visible tokens $\mathbf{X}_v$ into a vision transformer-based encoder. The learned representations for visible tokens $\mathbf{F}_v$ are concatenated with a fixed learnable representation f_m corresponding to the masked tokens. Subsequently, the positional information is added for both representations in the same order. These combined

tokens are then passed through a transformer-based decoder to estimate predictions $\hat{\mathbf{V}}$. The entire network is trained by reconstruction loss computed between ground-truth and predicted values for the masked tokens.

5.3.2 Trajectory-Aware Adaptive Token Sampler

We propose *TATS* module (g_θ) that can be easily integrated into the family of MAE (f_ϕ) architectures, can be trained from scratch and learns to sample motion-centric tokens without the use of any external pre-trained models to compute optical flow such as RAFT [Teed and Deng, 2020] in MGMAE [Huang et al., 2023], motion vectors in MGM [Fan et al., 2023] or having a motion-specific pre-training objective in MME [Sun et al., 2023]. In particular, we avail of TA [Patrick et al., 2021], which captures motion dynamics in a video by learning a probabilistic path of a token between frames. By exclusively sampling motion-centric tokens, *TATS* facilitates the encoder to learn more generic and expressive representations, which is crucial for downstream tasks such as action recognition. The computational overhead of *TATS* is minimal compared to MAE.

Trajectory Attention. In the *TATS* module, we apply TA [Patrick et al., 2021] across space-time tokens, where a trajectory represents the probabilistic path of a token in a video sequence determined by the motion between a pair of frames. A set of query-key-value vectors $\mathbf{q}_{st}, \mathbf{k}_{st}, \mathbf{v}_{st} \in \mathbb{R}^d$ is computed through linear projections (W) for a given space-time token $\mathbf{x}_{st}$ ($\mathbf{x}_{st} \in \mathbf{X}$) corresponding to space-time location st ('reference point') in a video. For $\mathbf{q}_{st}$, a set of trajectory tokens $\tilde{\mathbf{y}}_{stt'} \in \mathbb{R}^d$ is computed, encapsulating spatially pooled information weighted by the trajectory probability. These trajectory tokens extend throughout the video sequence and can be represented independently at different time steps t'.

$$\tilde{\mathbf{y}}_{stt'} = \sum_{s'} \mathbf{v}_{s't'} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{s't'} \rangle}{\sum_{\bar{s}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{\bar{s}t'} \rangle}. \tag{5.1}$$

Here, exp denotes the exponential function, $\langle ., . \rangle$ represents dot product, and $\cdot$ indicates element-wise multiplication. Next, trajectories are pooled across time to

capture intra-frame relationships. The trajectory tokens $\tilde{\mathbf{y}}_{stt'}$ are mapped to a new set of query, key, and value representations, denoted as $\tilde{\mathbf{q}}_{st}, \tilde{\mathbf{k}}_{stt'}, \tilde{\mathbf{v}}_{stt'}$, using the projection matrix $\tilde{W}$. Now $\tilde{\mathbf{q}}_{st}$ becomes the updated reference query for reference point st. $\tilde{\mathbf{q}}_{st}$ is then utilized to aggregate information across the temporal dimension using 1D attention given by:

$$\mathbf{y}_{st} = \sum_{t'} \tilde{\mathbf{v}}_{stt'} \cdot \frac{\exp\langle \tilde{\mathbf{q}}_{st}, \tilde{\mathbf{k}}_{stt'} \rangle}{\sum_{\bar{t}} \exp\langle \tilde{\mathbf{q}}_{st}, \tilde{\mathbf{k}}_{st\bar{t}} \rangle}. \tag{5.2}$$

The trajectory information is encoded in $\mathbf{y}_{st}$. In practice, we employ an approximation of TA (Orthoformer [Patrick et al., 2021]), which has linear complexity in space-time.

The **TATS module** (g_θ) has two branches, one of which processes the input tokens X by passing them through a block of TA followed by a *Linear* layer, and a *Softmax* activation to compute the probability scores $\pi_\theta(X) \in \mathbb{R}^N$ for all tokens.

$$\boldsymbol{Z} = \mathrm{TA}(\boldsymbol{X});\ \mathrm{Z} \in \mathbb{R}^{N \times d} \tag{5.3}$$

$$\pi_\theta(X) = \texttt{Softmax}(\texttt{Linear}(\boldsymbol{Z})) \in \mathbb{R}^N \tag{5.4}$$

Following AdaMAE [Bandara et al., 2023], these probability scores are utilized to define an N-dimensional categorical distribution over $\pi_\theta(X)$, from which visible token indices are sampled without replacement i.e. $\boldsymbol{I}_v \sim \mathrm{Categorical}(N, \pi_\theta(X))$. The masked token indices are the complement of visible token indices and are given by $\boldsymbol{I}_m = \overline{\boldsymbol{I}_v}$. The number of sampled visible tokens $\boldsymbol{N}_v = N \times (1 - \rho)$ and $\rho \in (0, 1)$ is the predefined mask ratio. This branch can be interpreted as the actor-network (or policy network), which outputs the probability of relevance for every token. In other words, this output probability can be perceived as policy π_θ(X) representing the likelihood of a token being selected given its token representation $\boldsymbol{X}$. The second branch processes the mean representation of all the tokens (X_μ) and passes it through a feed-forward network consisting of two linear layers with a

ReLU activation Linear(1568) $\rightarrow$ ReLU(Linear(784)) $\rightarrow$ 1. This can be interpreted as the value network, which learns to predict the expected reward for the current input tokens $\boldsymbol{X}$, given a mean state X_μ. We denote the output of the value network as $\psi_\theta(X_\mu)$. This value is used for computing the advantage $\boldsymbol{A}(X, I_m)$ as detailed in the optimization section. Overall, the computation of *TATS* (g_θ) can be represented as:

$$\pi_\theta(X), \psi_\theta(X_\mu) = g_\theta(X) \tag{5.5}$$

The complete architecture is shown in Fig 5.1.

5.3.3 Optimization

TATS can be conceptualized as an agent interacting with its environment, represented by the MAE, with the objective of learning an optimal masking strategy that removes redundant tokens while selecting only the most informative and motion-centric ones for encoding, given a mask ratio ρ. The environment provides feedback to *TATS* through a reward, which corresponds to the reconstruction error $\mathcal{L}_R$ [Bandara et al., 2023].

The intuition for this reward is that tokens with low reconstruction errors are easier to reconstruct and thus contain redundant information, whereas motion-centric tokens, which are more challenging to reconstruct, exhibit higher reconstruction error. Consequently, *TATS* must be optimized to prioritize the selection of these motion-centric tokens or tokens with higher reconstruction error. Our optimization strategy is loosely inspired from the application of RL [Goldberg, 2023] in the context of aligning large language model (LLM) outputs with human preferences. A major challenge in this formulation is the simultaneous training of both the agent (*TATS*) and the reward model (*MAE*), differing from conventional LLM approaches where the reward model is typically pre-trained separately based on human-labeled data. The joint optimization process incorporates two distinct losses, i.e. the reconstruction loss and the sampling loss, as outlined below.

Reconstruction Loss: To optimize the MAE (characterized by f_ϕ), we compute

the mean squared error loss $\mathcal{L}_R$ between the predicted and the normalized ground-truth RGB values of the masked tokens as shown in the following equation:

$$\mathcal{L}_R(\phi) = \frac{1}{N - N_v} \sum_{i \in \boldsymbol{I}_m} ||\tilde{\mathbf{V}}_i - \hat{\mathbf{V}}_i||_2 \tag{5.6}$$

where $\hat{\mathbf{V}}$ denotes the predicted tokens from the decoder, $\tilde{\mathbf{V}}$ represents the patch normalized ground-truth RGB values corresponding to the masked tokens.

Sampling Loss. *TATS* (g_θ) is optimized using the sampling loss $\mathcal{L}_S(\theta)$ based on PPO [Schulman et al., 2017]. To jointly train f_ϕ and g_θ from scratch, we propose a unified training approach that alternates between optimizing f_ϕ and g_θ. Initially, our objective is to train f_ϕ up to epoch m_o using random space-time masking, minimizing the reconstruction loss $\mathcal{L}_R$. This ensures that the MAE learns the task of reconstructing masked tokens, as the reconstruction error would be used as a reward for sampling the most challenging space-time tokens.

Since g_θ is trained using PPO, which requires episodes recorded from a previous state of g_θ. To facilitate this during Phase 1 (after m_o epochs), for every k steps, g_θ is kept frozen while $I_v \sim \text{Categorical}(N, \pi_{\theta_{\text{old}}}(X))$ and f_ϕ is optimized based on $\mathcal{L}_R(\phi)$. Simultaneously, the memory buffer $\mathcal{M}_b$ is updated with recorded episodes in the form of $\{X, \pi_{\theta_{\text{old}}}(I_m|X), \mathcal{L}_R(\phi).\text{detach}, \psi_{\theta_{\text{old}}}(X_\mu)\}$. Here, X represents the tokens, $\pi_{\theta_{\text{old}}}(I_m|X)$ denotes the probability of sampling the masked indices, $\mathcal{L}_R(\phi)$ corresponds to the reconstruction error from f_ϕ, and $\psi_{\theta_{\text{old}}}(X_\mu)$ represents the output of the value network. Using recorded rewards and value estimates, the advantage is computed as $A_{\theta_{\text{old}}}(\text{X}, I_m) = \mathcal{L}_R(\phi) - \psi_{\theta_{\text{old}}}(\text{X}_\mu)$.

In Phase 2, f_ϕ is frozen while g_θ is unfrozen. Recorded episodes are then sampled from $\mathcal{M}_b$, and the current state of g_θ is used for computing $\pi_\theta(X), \psi_\theta(X_\mu) = g_\theta(X)$ and $I_{v'} \sim \text{Categorical}(N, \pi_\theta(X))$. Notably, $\mathcal{L}_R(\phi)$ is detached from the computation graph to prevent gradient propagation in MAE during this step. The overall PPO objective used for training g_θ is defined by the following equation.

$$J^{PPO}(\theta) = \mathbb{E}\Big[c_1 J^{\text{CLIP}}(\theta) - c_2 \left(\psi_\theta(\mathrm{X}_\mu) - \mathcal{L}_R(\phi)\right)^2 + c_3 \mathbf{H}(\mathrm{X}, \pi_\theta)(.)\Big] \tag{5.7}$$

$$J^{\text{CLIP}}(\theta) = \mathbb{E}\Big[\min\left(r(\theta) A_{\theta_{\text{old}}}(\mathrm{X}, I_m), \text{clip}\left(r(\theta), 1-\epsilon, 1+\epsilon\right) A_{\theta_{\text{old}}}(\mathrm{X}, I_m)\right)\Big] \tag{5.8}$$

where $r(\theta) = \frac{\pi_\theta(I_{m'}|\mathrm{X})}{\pi_{\theta_{\text{old}}}(I_m|\mathrm{X})}$ represents the importance sampling ratio, and $\epsilon = 0.2$ is the clipping (clip) threshold. The term $(\psi_\theta(\mathrm{X}_\mu) - \mathcal{L}_R(\phi))^2$ serves as the objective for training the value network, representing the error in value estimation. $\mathbf{H}(\mathrm{X}, \pi_\theta)(.)$ denotes the entropy term associated with the tokens X and policy π_θ, promoting sufficient exploration. The coefficients c_1, c_2, c_3 balance $J^{\text{CLIP}}(\theta)$ (policy loss), value loss, and entropy term, respectively, in the overall PPO objective. After completing Phase 2, f_ϕ is unfrozen, $\mathcal{M}_b$ is reset, and the algorithm transitions back to Phase 1. This alternating process continues, switching between Phase 1 and Phase 2 iteratively throughout training. Since we want to minimize the sampling loss hence $\mathcal{L}_S(\theta) = -J^{PPO}(\theta)$. AdaMAE [Bandara et al., 2023] utilizes REINFORCE [Williams, 1992] which has high variance, however using PPO [Schulman et al., 2017] improves stability as it uses a clipped objective $J^{\text{CLIP}}(\theta)$ preventing it from making large updates, therefore balancing exploration and exploitation. Our training recipe is illustrated in Algorithm 1.

5.4 Experimental Setup

Datasets. We validate our method on four common and publicly-accessible benchmarks: SSv2 [Goyal et al., 2017], K400 [Kay et al., 2017], UCF101 [Soomro et al., 2012] and HMDB51 [Kuehne et al., 2011].

Data Preprocessing. Our data processing pipeline closely follows AdaMAE [Bandara et al., 2023] for pre-training. We extract 16 frames of dimension 224×224 from the videos, using a temporal stride of 4 (K400) and 2 (HMDB51/UCF101/SSv2), with the starting frame randomly selected [Feichtenhofer et al., 2022]. During pre-

Algorithm 1 Unified Training Recipe for joint optimization of MAE and TATS.

Require: Video V, MAE network f_ϕ, *TATS* module g_θ, mask ratio ρ, memory buffer $\mathcal{M}_b$, epochs E, Train only MAE epochs m_o, *TATS* update interval k, Total number of tokens N.

Initialize MAE f_ϕ and *TATS* g_θ.
for $e = 1$ to E **do**
  for step, batch in dataloader **do**
    tokenize V into X with indices $\{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_N\}$.
    if $e \leq m_o$ **then** ▷ Random Space-Time Masking Phase
      $I_v \sim$ Random Distribution with ρ
      optimize $\mathcal{L}_R = f_\phi(X_v)$ w.r.t. ϕ.
    else ▷ TATS Training Phase
      freeze g_θ, compute $\pi_{\theta_{\text{old}}}(X), \psi_{\theta_{\text{old}}}(X_\mu) = g_\theta(X)$.
      $I_v \sim \text{Categorical}(N, \pi_{\theta_{\text{old}}}(X))$ with ρ ; $I_m = \overline{I_v}$
      optimize $\mathcal{L}_R(\phi) = f_\phi(X_v)$ w.r.t. ϕ.
      episode $= \{X, \pi_{\theta_{\text{old}}}(I_m|X), \mathcal{L}_R(\phi).\text{detach}, \psi_{\theta_{\text{old}}}(X_\mu)\}$
      $\mathcal{M}_b$.update(episode)
      if step mod $k = 0$ **then** ▷ TATS Update
        freeze f_ϕ, unfreeze g_θ.
        for episode in $\mathcal{M}_b$ **do**
          compute $\pi_\theta(X), \psi_\theta(X_\mu) = g_\theta(X)$
          $I_{v'} \sim \text{Categorical}(N, \pi_\theta(X))$; $I_{m'} = \overline{I_{v'}}$
          optimize $\mathcal{L}_S(\theta) = -J^{PPO}(\theta)$ w.r.t. θ.
        end for
        unfreeze f_ϕ.
        $\mathcal{M}_b$.reset()
      end if
    end if
  end for
end for

training, we apply data augmentation techniques, including random resized cropping in the spatial domain, random scaling within the range $\in [0.5, 1]$, and random horizontal flipping [Feichtenhofer et al., 2022].

Implementation Details. We employ the ViT-Base model (≈87M parameters) [Dosovitskiy et al., 2020] for our experiments. The input video has the dimension $16 \times 3 \times 224 \times 224$ while the patch size is $2 \times 3 \times 16 \times 16$ (tubelet length = 2), yielding a total of 1568 tokens. Mask ratio ρ takes the value $\{0.85, 0.90, 0.95\}$. Our experiments contain two types of settings:

1. Small Scale Pre-training. For K400 and SSV2, we construct a smaller training data subset by sampling approximately 15% of the training set (equivalent to validation set size), while maintaining a class distribution consistent with the original dataset. Notably, the validation set remains unchanged from the original dataset. The standard train/validation sets for UCF101 and HMDB51 are used. The models have been pre-trained for 400 epochs with a batch size of 32 on 8 Nvidia A100 GPUs.

2. Large Scale Pre-training is also conducted on the full SSv2, however due to computational constraints, we only pretrain it for 400 epochs and $\rho = 0.95$ on 8 Nvidia A100 GPUs.

Evaluation on action recognition. To assess the effectiveness of the pre-trained encoder, we conduct end-to-end *fine-tuning* for the action recognition task over 100 epochs with the evaluation metric being top-1 and top-5 accuracy. Most of our experiments are conducted in a small-scale setting, while results for large-scale pre-training and fine-tuning are explicitly reported.

The code for reproducing the results presented in this Chapter is available at `https://github.com/rayush7/adaptive_vidmae`.

5.5 Results

We perform extensive quantitative and qualitative studies of our approach on the given datasets and compare the performance against [Bandara et al., 2023] and [Tong

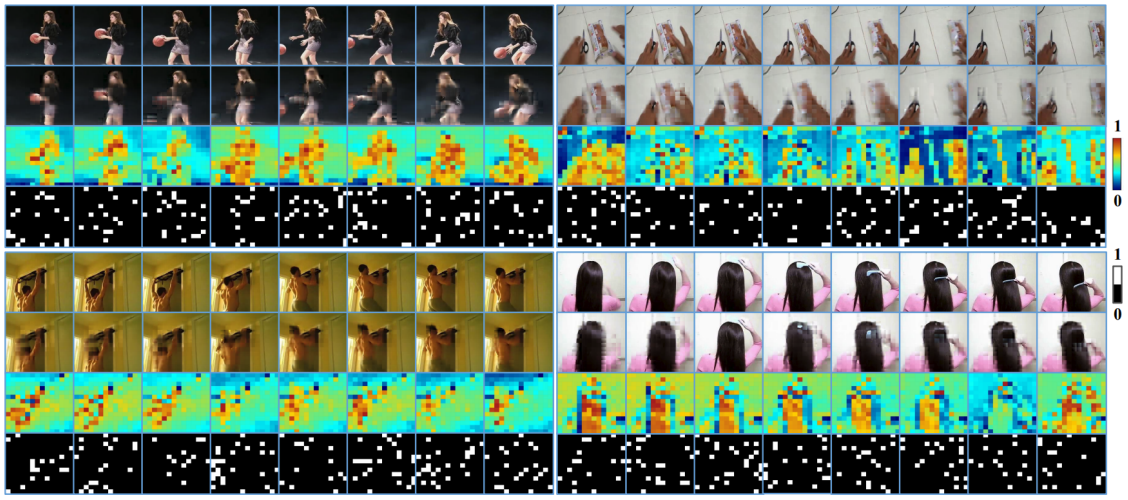


Figure 5.2: Visualization of adaptive masks learned by *TATS* for $\rho = 0.95$. The figure has four blocks: top-left (K400), top-right (SSv2), bottom-left (UCF101), and bottom-right (HMDB51). In each block, the first row shows video frames, the second presents predictions/reconstructions, the third depicts sampling probabilities for space-time tokens, and the fourth displays the learned adaptive binary masks.

et al., 2022] (baselines) respectively. For fair comparison with our method under small-scale pre-training setup, these baselines were also pretrained (finetuned) for 400 (100) epochs on the same subset (K400/SSv2) using their public source code and default configuration.

Fine-tuning Results. Table 5.1 presents the top-1 and top-5 accuracy obtained after fine-tuning our method across different mask ratios, $\rho = \{0.85, 0.90, 0.95\}$. Our approach consistently surpasses [Bandara et al., 2023, Tong et al., 2022] across all benchmark datasets and mask ratios with the exception of top-5 accuracy on HMDB51 with $\rho = 0.85$, which is marginally less than [Bandara et al., 2023]. Notably, even under an aggressive masking ratio ($\rho = 0.95$), our model demonstrates superior performance compared to these baselines. These results highlight the effectiveness and generalization capability of the proposed *TATS* module and the training strategy in terms of learning a better representation quality than learnt by [Bandara et al., 2023, Tong et al., 2022].

Transferability. Table 5.2 presents the transfer performance of our model on the action recognition task, pre-trained and fine-tuned across different datasets and mask ratio combinations. Our approach achieves better results than [Bandara et al.,

Table 5.1: Comparison of fine-tuning result of Our model against baselines ([Bandara et al., 2023, Tong et al., 2022]) on action recognition task across benchmark datasets and different ρ with top-1/top-5 accuracy as evaluation metric. ((↑) / (↓) : denotes increase/decrease in performance)

Dataset	Mask Ratio	VideoMAE [Tong et al., 2022]		AdaMAE [Bandara et al., 2023]		**Ours**	
	ρ	top-1	top-5	top-1	top-5	top-1	top-5
UCF101	0.85	80.36	94.95	83.98	96.37	85.94 (↑)	96.98 (↑)
	0.90	76.64	94.29	82.42	95.84	84.53 (↑)	96.37 (↑)
	0.95	65.86	89.14	80.83	95.26	81.75 (↑)	95.29 (↑)
HMDB51	0.85	40.82	71.61	41.28	73.37	41.60 (↑)	73.31 (↓)
	0.90	36.39	69.73	39.13	72.33	41.28 (↑)	73.76 (↑)
	0.95	33.98	65.36	37.70	70.38	38.67 (↑)	72.01 (↑)
Kinetics-400	0.85	42.26	68.28	38.97	64.68	43.24 (↑)	68.76 (↑)
	0.90	41.79	68.62	39.50	65.70	43.28 (↑)	68.85 (↑)
	0.95	39.73	66.15	39.42	65.14	41.70 (↑)	67.29 (↑)
SSv2	0.85	37.63	66.47	37.92	66.63	39.96 (↑)	68.10 (↑)
	0.90	37.85	66.86	38.10	66.29	40.79 (↑)	69.30 (↑)
	0.95	37.24	65.92	38.38	67.11	40.25 (↑)	68.73 (↑)

2023, Tong et al., 2022] across most settings, providing further insight into the strong transferability and generalization of our model.

Table 5.2: Comparison of transfer learning result of Our model against [Bandara et al., 2023, Tong et al., 2022] on action recognition across benchmark datasets and different ρ with top-1/top-5 accuracy as evaluation metric. ((↑) / (↓) / (–) : denotes increased/decreased/equivalent performance)

Dataset	Mask Ratio	VideoMAE [Tong et al., 2022]		AdaMAE [Bandara et al., 2023]		**Ours**	
From → To	ρ	top-1	top-5	top-1	top-5	top-1	top-5
Kinetics-400 → UCF101	0.85	84.91	96.51	85.49	96.93	86.94 (↑)	97.67 (↑)
	0.90	84.41	96.25	84.98	96.48	86.23 (↑)	97.27 (↑)
	0.95	82.40	95.80	84.03	96.50	85.17 (↑)	96.77 (↑)
Kinetics-400 → HMDB51	0.85	55.60	82.55	55.79	84.44	60.81 (↑)	84.44 (–)
	0.90	56.71	83.07	56.45	82.49	60.42 (↑)	83.59 (↑)
	0.95	53.26	79.75	54.10	81.25	58.14 (↑)	82.62 (↑)
Kinetics-400 → SSv2	0.85	36.42	65.50	36.72	65.72	38.39 (↑)	66.47 (↑)
	0.90	35.70	64.46	36.62	65.27	39.46 (↑)	67.25 (↑)
	0.95	34.11	62.64	36.88	65.64	38.13 (↑)	66.48 (↑)
SSv2 → UCF101	0.85	84.88	96.91	84.98	96.72	87.16 (↑)	97.38 (↑)
	0.90	83.88	96.75	84.64	97.06	86.81 (↑)	97.51 (↑)
	0.95	82.53	95.90	84.38	96.21	85.14 (↑)	97.11 (↑)
SSv2 → HMDB51	0.85	54.82	82.03	55.47	82.81	59.64 (↑)	84.83 (↑)
	0.90	55.92	83.40	55.86	84.31	60.35 (↑)	85.42 (↑)
	0.95	52.41	80.14	54.69	84.18	58.40 (↑)	83.59 (↓)

Qualitative Assessment. We conduct a qualitative analysis by visualizing the learned adaptive binary masks learned by the *TATS* module across the benchmark datasets and different ρ, as shown in Figure 5.2. We observe that *TATS* learns to sample motion-centric tokens while also undergoing sufficient exploration enabling better generalization. Additionally, we visualize the learned TA across all space-time patches by averaging all heads, as depicted in Figure 5.3. It is quite evident that our *TATS* module accurately models motion trajectories of the space-time tokens as

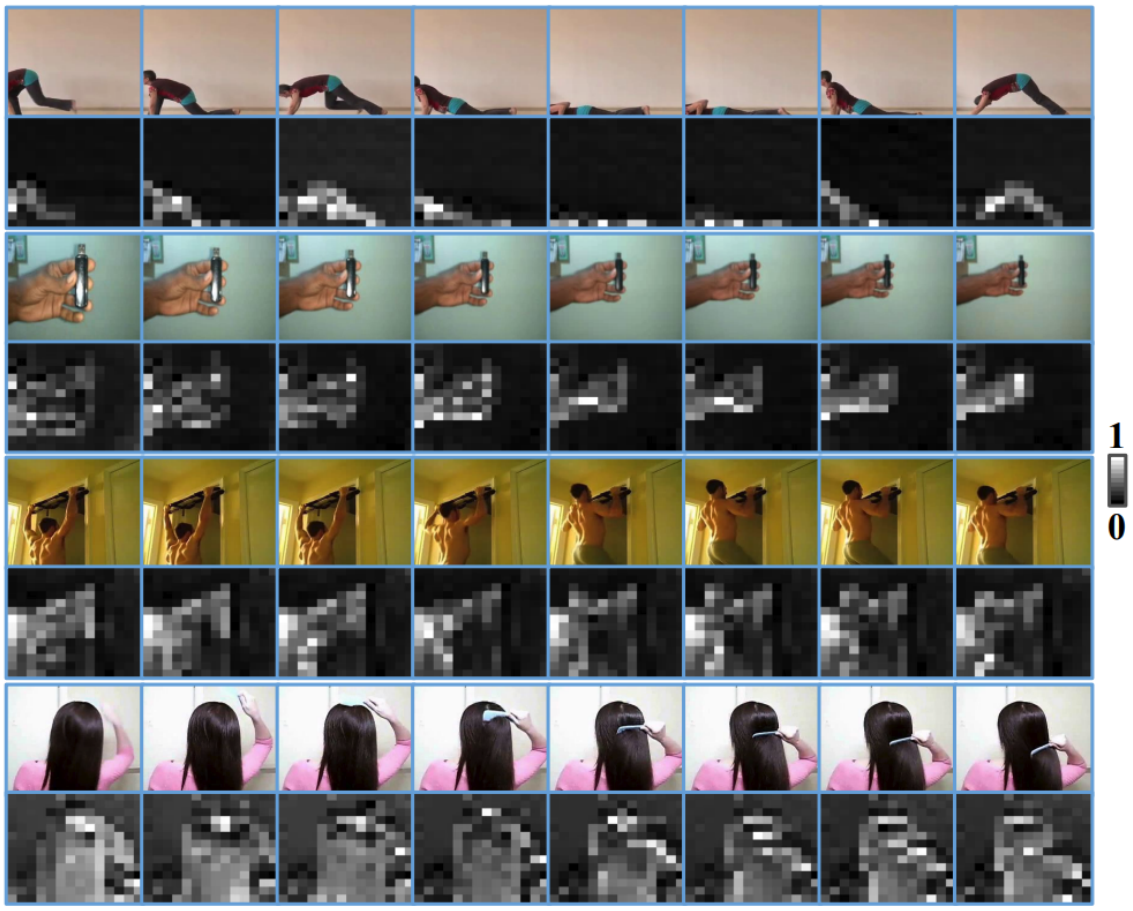


Figure 5.3: Visualization of the TA learnt by *TATS*. The figure comprises four blocks : K400, SSv2, UCF101, and HMDB51 in top to bottom order. In each block, the first row shows video frames, the second depicts the trajectory attention on space-time tokens averaged across different heads.

they evolve over time in the video, thereby enabling the sampling of motion-centric space-time patches. This also validates the formulation of the $\mathcal{L}_s$ and the training recipe to jointly train MAE and *TATS*.

Table 5.3: **Large Scale Pre-training and Finetuning Results.** Comparison of finetuning result of Our model against baselines ([Bandara et al., 2023, Tong et al., 2022]) on action recognition task for full SSv2 and $\rho = 0.95$ with top-1/top-5 accuracy as evaluation metric. ((↑) : denotes increase in performance)

Method	top-1	top-5
VideoMAE [Tong et al., 2022]$_{\rho=95\%}$	59.38	84.17
AdaMAE [Bandara et al., 2023]$_{\rho=95\%}$	63.06	85.89
Ours$_{\rho=95\%}$	65.82 (↑)	88.50(↑)

Large Scale Pre-training Results. We conduct pre-training (400 epochs) and finetuning (100 epochs) of our model on the full SSv2 [Goyal et al., 2017] dataset for

$\rho = 0.95$ on 8 Nvidia A100 GPUs. In order to ensure fairness in comparison, we also pre-train (400 epochs) and finetune (100 epochs) both baselines VideoMAE [Tong et al., 2022] and AdaMAE [Bandara et al., 2023] on the full SSv2 for $\rho = 0.95$ with the same GPU setup using their public source code and default configuration.

Table 5.3 presents the top-1 and top-5 accuracy obtained in this experiment. We observe that our approach outperforms both the baselines under aggressive masking setting even for large scale experiments. This highlights the effectiveness and generalization capability of the proposed *TATS* module and the training strategy in terms of learning a better feature quality than learnt by [Bandara et al., 2023, Tong et al., 2022]. Due to the availability of limited computational resources, our experiments in this setup are limited.

5.6 Additional Implementation Details

5.6.1 Hyper-parameter Setting

Pre-training. The hyperparameter configurations used during the pre-training phase across all benchmark datasets are presented in Table 5.4. For (m_o, k), hyperparameter tuning is conducted on the UCF101 and HMDB51 datasets (Table 5.5), and the configuration that minimizes the reconstruction error is selected. Similarly we also perform hyperparameter tuning for coefficients (c_1,c_2,c_3) in Table 5.6 during pretraining on UCF101 and observe that (1e-4, 1e-4, 1e-4) minimizes the reconstruction error. Empirical observations indicate that the optimal configuration for UCF101 also performs effectively on subset of K400 and SSv2 (small scale pre-training setup). It is to be noted that we use reconstruction loss for tuning these hyper-parameters because behaviour of reconstruction loss during pretraining is more interpretable in terms of convergence than the sampling loss.

Fine-tuning. The hyperparameter setting for end-to-end fine-tuning on the downstream task of action recognition across all benchmarks is summarized in Table 5.7.

Table 5.4: Hyperparameter setting for pre-training across all benchmark datasets.

Configuration	Value
Learning rate for g_θ - lp	1.5e-6
Epochs to train f_ϕ only - m_o	10
Steps to train f_ϕ and record g_θ episodes - k	1
Softmax Temperature	1
Policy loss coefficient - c_1	1e-4
Value loss coefficient - c_2	1e-4
Entropy coefficient - c_3	1e-4
Optimizer	AdamW
Optimizer betas	0.9, 0.95
Batch size	32
Base learning rate	1.5e-4
Learning rate schedule	cosine decay
Warmup epochs	40
Augmentation	MultiScaleCrop

Table 5.5: Hyperparameter (m_o, k) tuning for pre-training, evaluated based on reconstruction error on UCF101 and HMDB51. Same configuration is adopted for SSv2 and K400 as in UCF101.

(m_o, k)	UCF101	HMDB51
(0, 1)	0.5211	0.8051
(1, 1)	0.5205	0.8195
(5, 1)	0.5304	0.8535
(10, 1)	0.5135	0.8278
(25, 1)	0.5269	0.8987
(100, 1)	0.6662	0.9291
(50, 5)	0.7735	0.9772
(50, 10)	0.8149	0.9776
(50, 25)	0.9201	-

Table 5.6: Hyperparameter (c_1, c_2, c_3) tuning for pre-training, evaluated based on reconstruction error on UCF101. Same configuration is adopted for SSv2, K400 and HMDB51. (m_o, k) are fixed as $(10, 1)$

(c_1, c_2, c_3)	UCF101
(1e-4, 1e-3, 1e-3)	0.5188
(1e-4, 1e-3, 1e-4)	0.5167
(1e-4, 1e-4, 1e-3)	0.5246
(1e-3, 1e-4, 1e-4)	0.8482
(1e-4, 1e-4, 1e-4)	0.5135
(1e-5, 1e-4, 1e-4)	0.5239
(1e-3,1e-3,1e-4)	0.5215
(1e-3, 1e-3, 1e-4)	0.7869
(1e-5, 1e-5, 1e-5)	0.5173

Table 5.7: Hyperparameter setting for end-to-end fine-tuning for all benchmark datasets.

Configuration	Value
Optimizer	AdamW
Optimizer Betas	{0.9, 0.999}
Batch size	8
Weight Decay	5e-2
Base Learning Rate	1e-3
Learning Rate Schedule	cosine decay
Layer-wise learning rate decay	0.75
Warmup epochs	5
RandAug	9, 0.5
Label Smoothing	0.1
Mixup	0.8
DropPath	0.1
# Temporal Clips	5 (k400), 2 (ssv2/hmdb/ucf)
# Spatial Crops	3

Table 5.8: Encoder-Decoder architecture based on AdaMAE [Bandara et al., 2023]. TATS : Trajectory Aware Adaptive Token Sampler. MHA : Multi-Head Self-Attention

Stage	ViT-Base	Output shape
Input Video	stride $4 \times 1 \times 1$ for K400 stride $2 \times 1 \times 1$ for ssv2/ucf/hmdb	$3 \times 16 \times 224 \times 224$
Tokenization	stride $2 \times 16 \times 16$ emb. dim 768 kernel size $2 \times 16 \times 16$	1568×768
Masking	TATS Masking mask ratio ρ	$[(1-\rho) \times 1568] \times 768$
Encoder	$[MHA(768)] \times 12$	$[(1-\rho) \times 1568] \times 768$
Projection	$MHA(384)$ concat masked tokens	1568×384
Decoder	$[MHA(384)] \times 4$	$[(1-\rho) \times 1568] \times 384$
Projector	$MLP(1536)$	1568×1536
Reshaping	from 1536 to $3 \times 2 \times 16 \times 16$	$3 \times 16 \times 224 \times 224$

5.6.2 Encoder-Decoder Architecture

We adopt an asymmetric encoder-decoder architecture [Bandara et al., 2023] for self-supervised pre-training and augment it with *TATS* module and only keep the encoder during the fine-tuning. In particular, the design of the encoder-decoder is

based on 16-frame vanilla ViT-Base architecure. Table 5.8 provides an overview of the encoder-decoder architecture utilized in our framework.

5.7 Ablation Studies

We carry out an ablation study on UCF101 using models pre-trained with $\rho = 0.95$ for 400 epochs and fine-tuned on the action recognition task for 100 epochs. The ablation results are illustrated in Table 5.9.

1. Effect of Trajectory Attention. In Table 5.9a, we analyze the effect of integrating TA within the *TATS* module compared to the Multi-Head Self-Attention (MHA). Our findings indicate that TA achieves a top-1 accuracy of 81.75% while utilizing 25.36 GB of memory, outperforming MHA. This highlights the efficiency of TA in delivering superior performance with reduced memory consumption. Furthermore, our results also validate that TA effectively captures motion trajectories in a self-supervised manner, without relying on any motion-specific learning objective.

2. Effect of Decoder Depth. Table 5.9b examines the impact of different decoder depths, specifically the number (#) of transformer blocks in the decoder's architecture. Our findings show that the best performance is achieved with # Blocks = 1, yielding a top-1 accuracy of 81.75%. This observation aligns with the results observed in [Bandara et al., 2023, Tong et al., 2022].

3. Effect of Reconstruction Loss Function. In Table 5.9e, we examine the effect of the reconstruction objective, specifically comparing L1 and MSE losses. Following the standard approach introduced in VideoMAE [Tong et al., 2022], we also explore computing these losses (L1/MSE) using both raw pixel values and per-patch normalized pixels. Our results indicate that MSE loss with per-patch normalization achieves the highest top-1 accuracy of 81.75%.

4. Effect of Number of Trajectory Attention Blocks. In Table 5.9d, we investigate the effect of varying the # of TA blocks in *TATS*. Our results indicate that the configuration with # TA Blocks = 1 yields the highest top-1 accuracy of 81.75%. As we increase the # TA Blocks, the performance decreases while the

case	ratio	top-1	top-5	memory
MHA	0.95	81.59	95.29	25.37 GB
TA	**0.95**	**81.75**	**95.42**	**25.36 GB**

(a) **Effect of Trajectory Attention**. Better performance is obtained with TA with marginally less memory usage.

blocks	top-1	top-5	memory
1	81.46	95.07	16.54 GB
2	80.83	95.02	19.48 GB
4	**81.75**	**95.29**	**25.36 GB**
8	79.10	94.68	37.13 GB

(b) **Different decoder depth**. Our method performs best when # of decoder blocks = 4.

method	memory	top-1
VideoMAE	20.94 GB	65.86
AdaMAE	26.17 GB	80.83
Ours	**25.36 GB**	**81.75**

(c) **Memory Usage**. Our method uses less memory (pretraining) than AdaMAE [Bandara et al., 2023] while achieving significantly higher performance (finetuning) than VideoMAE [Tong et al., 2022].

case	top-1	top-5	memory
TA (# Blocks = 1)	**81.75**	**95.29**	**25.36 GB**
TA (# Blocks = 2)	65.17	88.59	32.18 GB
TA (# Blocks = 3)	67.35	90.20	39.00 GB

(d) **Number of TA blocks in *TATS***. Our method performs best when # of TA blocks = 1.

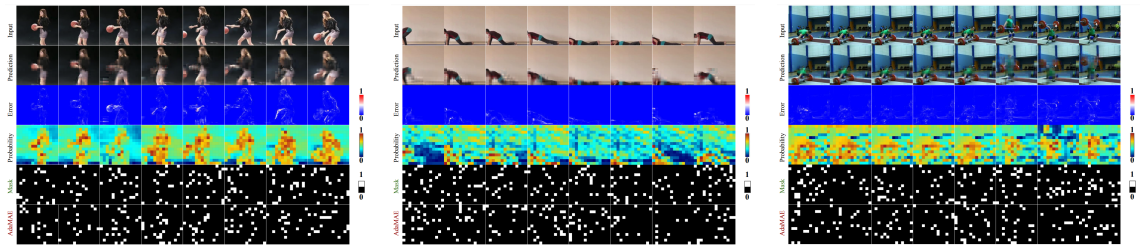
case	top-1	top-5
L1 loss (w norm.)	81.51	95.58
L1 loss (w/o norm.)	81.41	95.02
MSE loss (w norm.)	**81.75**	**95.29**
MSE loss (w/o norm.)	81.61	95.14

(e) **Reconstruction Loss function**. The best result is obtained by optimizing MSE loss with local patch normalization.

Table 5.9: Ablation analysis is conducted on the UCF101 dataset using models pretrained with mask ratio $\rho = 0.95$ for 400 epochs and fine-tuned on action recognition task for 100 epochs. The default choice of our method is highlighted in **gray** color.

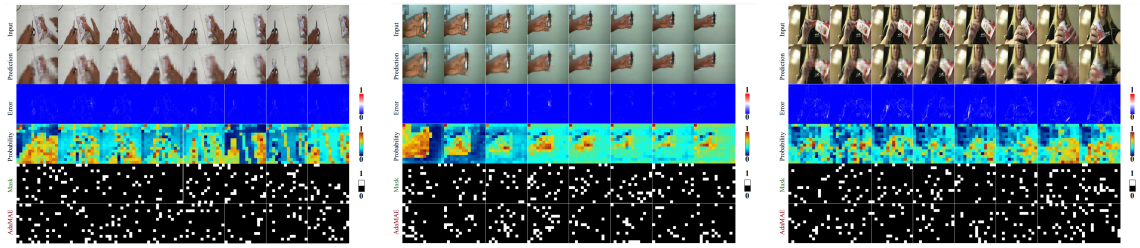
memory usage increases.

5. Memory Usage. In Table 5.9c, we inspect the memory usage of our approach in comparison to AdaMAE [Bandara et al., 2023] and VideoMAE [Tong et al., 2022]. Our method demonstrates lower memory consumption (pretraining) and better performance (finetuning) than AdaMAE [Bandara et al., 2023]. Although VideoMAE [Tong et al., 2022] utilizes less memory (pretraining) than our approach, our method significantly outperforms it in terms of top-1 accuracy (finetuning) on UCF101, achieving 81.75% compared to only 65.86% by VideoMAE [Tong et al., 2022].



(a) $\rho = 0.95$ (b) $\rho = 0.9$ (c) $\rho = 0.85$

Figure 5.4: Sample visualizations of a Kinetics 400 video using adaptive sampling with *TATS* at different mask ratios. Comparison shown with AdaMAE [Bandara et al., 2023] masks.



(a) $\rho = 0.95$ (b) $\rho = 0.9$ (c) $\rho = 0.85$

Figure 5.5: Sample visualizations of a SSv2 video using adaptive sampling with *TATS* at different mask ratios. Comparisons are shown with AdaMAE [Bandara et al., 2023] masks.

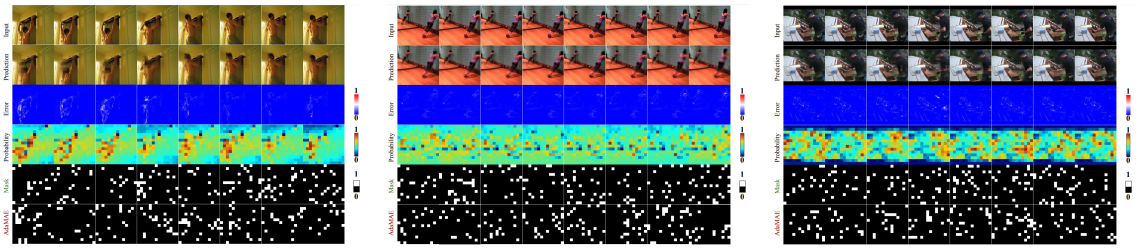
5.8 Mask Visualization

Here we show visualizations adaptive sampling learned by our *TATS* module across benchmark dataset for different mask ratios $\rho = \{0.95, 0.9, 0.85\}$ in Figure 5.4a, 5.4b, 5.4c, 5.5a, 5.5b, 5.5c, 5.6a, 5.6b, 5.6c, 5.7a, 5.7b, 5.7c.

In all of these Figures, first row represents input video frames, the second row depicts the prediction/reconstruction, the third row shows the reconstruction error, the fourth row represents the probability of sampling the space-time patch, fifth row shows the adaptive masks learned by *TATS*. The last row depicts the binary masks learned by AdaMAE [Bandara et al., 2023] for comparison.

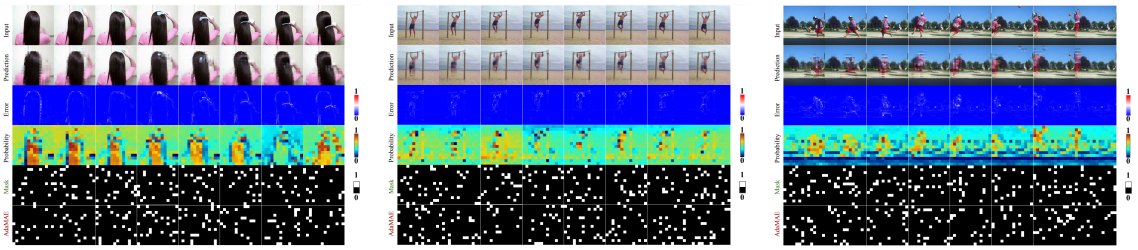
5.9 Conclusions

This Chapter addresses the hypothesis $\mathbf{H_3}$ (Chapter 1) which presumes that incorporating adaptive computation strategies into the self-supervised training objective



(a) $\rho = 0.95$ (b) $\rho = 0.9$ (c) $\rho = 0.85$

Figure 5.6: Sample visualizations of a UCF101 video using adaptive sampling with *TATS* at different mask ratios. Comparisons are made with AdaMAE [Bandara et al., 2023] masks.



(a) $\rho = 0.95$ (b) $\rho = 0.9$ (c) $\rho = 0.85$

Figure 5.7: Sample visualizations of a HMDB51 video using adaptive sampling with *TATS* at different mask ratios. Compared against AdaMAE [Bandara et al., 2023] masks.

enables learning of more transferable and generalizable video representations in a more *efficient* manner compared to the static computation. In particular, the objective of this chapter is to evaluate $\mathbf{R_5}$: *How can we incorporate adaptive computation in a self-supervised pre-training objective such as MVM to dynamically select informative space-time tokens based on the given input?* and $\mathbf{R_6}$: *Are representations learnt through dynamic computation (adaptive masking) as transferable to downstream tasks (action recognition) as the ones learnt with static computation (random masking)?*. The downstream task selected for this study is the task of action recognition (Chapter 2,5).

To summarise, in this Chapter we propose a novel and generalizable *TATS* module that enhances MAE pre-training for videos by adaptively selecting motion-centric tokens based on their spatio-temporal motion trajectories. *TATS* can be integrated into the MAE framework without requiring additional modalities like optical flow (e.g., RAFT [Teed and Deng, 2020]) or external pre-trained models such as DI-

NOv2 [Oquab et al., 2023] or CLIP [Radford et al., 2021] for motion priors or semantic cues. We also introduce a unified training framework (Algorithm 1) that enables the joint optimization of MAE and *TATS* from scratch using PPO [Schulman et al., 2017], enhancing stability during pre-training even under aggressive masking (**answering $\mathbf{R_5}$**). Finally, we perform an extensive quantitative, qualitative and ablation assessment (Tables 5.1,5.2,5.9) on benchmark datasets (K400, SSv2, UCF101, HMDB51) for the downstream task of action recognition, showcasing the effectiveness, generalization, transferability, and efficiency of our approach compared to state-of-the-art methods (**answering $\mathbf{R_6}$**).

However, there are few limitations with the work presented in this Chapter. Our proposed *TATS* and training recipe does need to be empirically validated on other downstream tasks and extended to other modalities. Furthermore, with the recent resurgence in RL research due to its applications in LLMs, it is important to reconsider strategies that integrate dynamic computation into masked modeling approaches, optimizing them through RL algorithms. We plan to conduct future studies around these topics.

Chapter 6

Conclusion

In this chapter, we bring together all the key findings of our research to highlight our contributions and discuss their significance. We review the research questions that we set out in Chapter 1 and assess how well we have answered them. In this way, we summarize the important insights from our work, discuss practical implications, and suggest directions for future research. By doing so, we aim to provide a clear and concise conclusion highlighting the value of this work.

6.1 Answers to Research Questions

R_1 : How can we leverage the power of SSL to capture spatio-temporal diversities and relationships involved in videos?

In Chapter 3, we presented a self-supervised model that can be pre-trained for the GEBD task (Figure 3.1). The GEBD task is an ideal problem for self-supervised learning, given that the task aims to learn generic boundaries and is not biased towards any predefined action categories from pre-trained state-of-the-art action recognition models. In order to learn spatial diversity, fine-grained temporal coherence and long-range temporal dependencies we reformulated the SSL objective at frame-level and clip-level to learn effective and structured video representations. Through our extensive evaluation, we achieved comparable performance to self-supervised state-of-the-art methods on the Kinetics-GEBD and TAPOS as shown

in Table 3.2 and Table 3.3, respectively. This demonstrates that by designing relevant self-supervised pretext tasks, it is indeed plausible to embed spatial diversity, fine-grained temporal coherence, and long-range temporal dependencies into the learned model.

$\mathbf{R_2}$: How can we develop an SSL framework for video understanding that accounts for both appearance and motion features? Do we need an explicit motion-specific training objective, or can this be implicitly achieved?

In Chapter 3, we augmented our ResNet-50 encoder with a differentiable motion learning *MotionSqueeze* module and observed that the augmented encoder can capture motion patterns (Table 3.4) on the fly despite being trained from scratch and without any motion specific self-supervised objective. Furthermore, this augmented encoder further complements the overall performance on the downstream GEBD task as highlighted in Table 3.2 and Table 3.3. Additionally, the motion features learnt are generic since the model is only pre-trained on Kinetics-GEBD but generalizes to the TAPOS dataset as well, as shown in Figure 3.3. This indicates that augmenting the encoder with a motion learning module allows for the implicit learning of motion priors, even in a self-supervised setting and without a motion-specific pretext task. Overall, this enables the learning of motion patterns in conjunction with appearance features from the augmented encoder.

$\mathbf{R_3}$: Is it possible to synthetically generate generic PAs by introducing spatio-temporal distortions into normal data in order to detect real-world anomalies effectively?, and importantly, can such PAs transfer across multiple VAD datasets?

In Chapter 4, we presented a novel and generic spatio-temporal PAs generator vital for VAD tasks without incorporating strong inductive biases. We achieve this by adding perturbation in the frames of normal videos by inpainting a masked out region using a pre-trained LDM and by distorting optical flow by applying mixup-like augmentation (Figure 4.2a). The observations made in (Table 4.4, Table 4.3,

Table 4.5) show that our PAs generalise and enable the detection of real-world anomalies through PAs. Furthermore, extensive experiments also validate the transferability and interpretability aspects of our PAs across benchmark VAD datasets. This implies that our synthetically generated spatio-temporal PAs facilitate real-world anomaly detection and are also transferable across datasets.

R_4 : How can we design a VAD pipeline that aggregates different anomaly indicators to create a unified anomaly scoring mechanism that effectively captures spatial, temporal, and semantic inconsistencies?

In Chapter 4, we introduced a simple unified VAD framework that learns three types of anomaly indicators, i.e. reconstruction quality, temporal irregularity and semantic inconsistency in an OCC setting (Figure 4.1). Extensive evaluation shows that our framework, achieves comparable performance to other SOTA reconstruction methods and PA generators with predefined assumptions across multiple datasets (Table 4.4, 4.3) without any end-to-end finetuning or any post-processing. This indicates the effectiveness and generalisation of our PAs and VAD pipeline.

R_5 : How can we incorporate adaptive computation in a self-supervised pre-training objective such as MVM to dynamically select informative space-time tokens based on the given input?

In Chapter 5, we proposed a novel and generalizable *TATS* module that enhances MAE pre-training for videos by adaptively selecting motion-centric tokens based on their spatio-temporal motion trajectories (Figure 5.1). *TATS* can be integrated into the MAE framework without requiring additional modalities like optical flow (e.g., RAFT [Teed and Deng, 2020]) or external pre-trained models such as DINOv2 [Oquab et al., 2023] or CLIP [Radford et al., 2021] for motion priors or semantic cues. We also introduced a unified training framework (Algorithm 1) that enables the joint optimization of MAE and *TATS* from scratch using PPO [Schulman et al., 2017], enhancing stability during pre-training even under aggressive masking. Additionally, we performed an extensive quantitative, qualitative and ablation assessment (Tables 5.1,5.9) on benchmark datasets (K400, SSv2, UCF101, HMDB51)

for the downstream task of action recognition, showcasing the effectiveness, generalization, transferability, and efficiency of our approach compared to state-of-the-art methods. This implies that integrating adaptive computation into the masked video modeling framework and jointly training for both the MAE and adaptive computation objectives enables the dynamic sampling of the most informative space-time tokens based on the input.

$\mathbf{R_6}$: Are representations learnt through dynamic computation (adaptive masking) as transferable to downstream tasks (action recognition) as the ones learnt with static computation (random masking)?

In Chapter 5, we compared the performance of our proposed method with the random tube masking strategy (STMAE [Tong et al., 2022]), a predefined (static computation) masking approach, on the downstream task of action recognition. As shown in Table 5.2, our adaptive token sampling strategy clearly outperforms static computation methods in transfer performance. This indicates that adaptive computation strategies are more effective than static computation techniques in harnessing the expressivity and capabilities of encoders.

6.2 Research Contributions

The per-chapter research contributions can be summarised as follows:

Chapter 3

1. We revisited and extended a simple self-supervised method VCLR [Kuang et al., 2021] by modifying its pretext tasks by splitting them into frame-level and clip-level to learn effective video representations (cVCLR). We further augmented the encoder with a differentiable motion feature learning module for GEBD.

2. We conducted exhaustive evaluation on the Kinetics-GEBD and TAPOS datasets and showed that our approach achieves comparable performance to the self-supervised state-of-the-art methods without using enhancements like model

ensembles, pseudo-labeling or the need for other modality features (e.g. audio).

3. We showed that the model can learn motion features under self-supervision even without having any explicit motion-specific pretext task.

Chapter 4

1. We proposed a novel and generic spatio-temporal pseudo-anomaly generator for VAD encompassing inpainting of a masked out region in frames using an LDM and applying mixup augmentation to distort the optical flow.

2. We introduced a simple unified VAD framework that measures and aggregates three different indicators of anomalous behaviour, namely reconstruction quality, temporal irregularity and semantic inconsistency in an OCC setting.

3. Extensive experiments on *Ped2, Avenue, ShanghaiTech* and *UBnormal* showed that our method achieves comparable performance to other existing SOTA PAs generation and reconstruction based methods under the OCC setting (Table 4.4, 4.3) without any end-to-end finetuning or any post-processing. This validates that our method is a generic video anomaly detector and our spatio-temporal PAs generation process is transferable across multiple datasets.

Chapter 5

1. We proposed a novel and generalizable TATS module that learns to adaptively sample motion-centric tokens for MAE pre-training by modeling their motion trajectories in videos. TATS can be seamlessly integrated into the MAE framework and does not rely on auxiliary modalities like optical flow (RAFT [Teed and Deng, 2020]) or external pre-trained models (DINOv2 [Oquab et al., 2023], CLIP [Radford et al., 2021]) for motion or semantic cues.

2. Additionally, we introduced a unified training recipe (Algorithm 1) that facilitates the joint optimization of both MAE and TATS from scratch using

PPO [Schulman et al., 2017] to ensure stable convergence during pre-training even with aggressive masking.

3. Finally, we conducted a comprehensive evaluation on four benchmark datasets (K400, SSv2, UCF101, HMDB51) for action recognition to demonstrate the effectiveness, generalization, transferability, and efficiency of our work compared to the state-of-the-art methods (Tables 5.1,5.2,5.9).

The code for reproducing the results presented in this thesis can be found at `https://github.com/rayush7?tab=repositories`.

6.3 Perspectives for Future Work

Notwithstanding the advances made by the research reported in this thesis on the topic of video understanding, there are exciting opportunities for future work. We document some of these in the following.

- In Chapter 3, our approach does not avail of more powerful models, e.g. transformers as in [Li et al., 2022a], or cascaded networks as in [Hong et al., 2021]. Additionally, since the MS module is directly applied on feature maps, it learns global motion features. However, in GEBD, the boundaries are generic and every type of motion may not indicate a boundary, hence a more fine-grained motion module can boost the performance. Finally, due to computational constraints, our self-supervised model is only pre-trained on the Kinetics-GEBD dataset; however, pre-training the model on Kinetics-400 could yield even better performance on the downstream GEBD task.

- In Chapter 4, our model was not trained in an end-to-end fashion and does not avail of more powerful architectures (vision transformers or 3D-ResNets) due to limited computational resources, which might boost the performance. It will also be interesting to make this setting adaptive by learning a policy network to select which anomaly indicator among poor reconstruction quality,

temporal irregularity, and semantic inconsistency contributes more towards the detection of real-world anomalies. Also, the notion of generating latent space PAs for VAD through LDMs or manifold mixup remains to be investigated.

- In Chapter 5, our proposed *TATS* module and training recipe needs to be empirically validated on other downstream tasks and extended to other modalities. Furthermore, with the recent resurgence in RL research due to its applications in LLMs, it is important to reconsider strategies that integrate dynamic computation into masked modeling approaches, optimizing them through RL algorithms. We plan to conduct future studies around these topics.

6.4 Closing Remarks

In this thesis, we addressed several underexplored aspects of video representation learning: What makes a good video representation? What properties should a video representation have? Specifically, we focused on learning *structured, robust, and efficient representations* in a self-supervised setting. This thesis offers several key insights. Firstly, a structured video representation can be learned by designing video-specific self-supervised pretext tasks that capture fine-grained (temporally granular) and global (temporally persistent) features while also leveraging motion patterns. Secondly, video representations can be learned to be more resilient and robust to spatio-temporal perturbations, such as lighting variations, background noise, and clutter, while ensuring strong attentiveness to the relevant information (action, motion patterns, human-object interaction) within the videos. Lastly, video data contains a high degree of redundancy that must be filtered out while retaining the essential information. Incorporating adaptive computation strategies into self-supervised representation learning techniques, such as masked video modeling, can enhance the learning of more transferable and generalizable features in an efficient way. We hope this research can motivate further research in the direction of representation learning for video understanding.

Bibliography

[Abati et al., 2019] Abati, D., Porrello, A., Calderara, S., and Cucchiara, R. (2019). Latent space autoregression for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 481–490.

[Abu Farha et al., 2018] Abu Farha, Y., Richard, A., and Gall, J. (2018). When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 5343–5352.

[Acsintoae et al., 2022] Acsintoae, A., Florescu, A., Georgescu, M.-I., Mare, T., Sumedrea, P., Ionescu, R. T., Khan, F. S., and Shah, M. (2022). 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.

[Agrawal et al., 2015] Agrawal, P., Carreira, J., and Malik, J. (2015). Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pages 37–45.

[Ahsan et al., 2019] Ahsan, U., Madhok, R., and Essa, I. (2019). Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE.

[Aich et al., 2023] Aich, A., Peng, K.-C., and Roy-Chowdhury, A. K. (2023). Cross-domain video anomaly detection without target domain adaptation. In *Proceed-*

ings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2579–2591.

[Akiva et al., 2023] Akiva, P., Huang, J., Liang, K. J., Kovvuri, R., Chen, X., Feiszli, M., Dana, K., and Hassner, T. (2023). Self-supervised object detection from egocentric videos. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5202–5214.

[Alayrac et al., 2022] Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (2022). Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

[Amrani et al., 2021] Amrani, E., Ben-Ari, R., Rotman, D., and Bronstein, A. (2021). Noise estimation using density estimation for self-supervised multimodal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6644–6652.

[Arnab et al., 2021a] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021a). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846.

[Arnab et al., 2021b] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021b). ViViT: A Video Vision Transformer. arXiv:2103.15691 [cs].

[Asano et al., 2019] Asano, Y. M., Rupprecht, C., and Vedaldi, A. (2019). Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*.

[Astrid et al., 2021a] Astrid, M., Zaheer, M. Z., Lee, J.-Y., and Lee, S.-I. (2021a). Learning not to reconstruct anomalies. In *BMVC*.

[Astrid et al., 2021b] Astrid, M., Zaheer, M. Z., and Lee, S.-I. (2021b). Synthetic temporal anomaly guided end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 207–214.

[Aydemir et al., 2023] Aydemir, G., Xie, W., and Güney, F. (2023). Self-supervised Object-centric Learning for Videos. In *Advances in Neural Information Processing Systems*.

[Ba, 2016] Ba, J. L. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

[Bai et al., 2020] Bai, Y., Fan, H., Misra, I., Venkatesh, G., Lu, Y., Zhou, Y., Yu, Q., Chandra, V., and Yuille, A. (2020). Can temporal information help with contrastive self-supervised learning? *arXiv preprint arXiv:2011.13046*.

[Bandara et al., 2023] Bandara, W. G. C., Patel, N., Gholami, A., Nikkhah, M., Agrawal, M., and Patel, V. M. (2023). Adamae: Adaptive masking for efficient spatiotemporal learning with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14507–14517.

[Banerjee et al., 2005] Banerjee, A., Dhillon, I. S., Ghosh, J., Sra, S., and Ridgeway, G. (2005). Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9).

[Bao et al., 2022] Bao, H., Dong, L., Piao, S., and Wei, F. (2022). Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*.

[Bardes et al., 2021] Bardes, A., Ponce, J., and LeCun, Y. (2021). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*.

[Beltagy et al., 2020] Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[Bengio et al., 2011] Bengio, Y., Bastien, F., Bergeron, A., Boulanger–Lewandowski, N., Breuel, T., Chherawala, Y., Cisse, M., Côté, M., Erhan, D., Eustache, J., Glorot, X., Muller, X., Pannetier Lebeuf, S.,

Pascanu, R., Rifai, S., Savard, F., and Sicard, G. (2011). Deep learners benefit more from out-of-distribution examples. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 164–172, Fort Lauderdale, FL, USA. PMLR.

[Bengio et al., 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

[Bertasius et al., 2021a] Bertasius, G., Wang, H., and Torresani, L. (2021a). Is space-time attention all you need for video understanding? In *ICML*, page 4.

[Bertasius et al., 2021b] Bertasius, G., Wang, H., and Torresani, L. (2021b). Is Space-Time Attention All You Need for Video Understanding? arXiv:2102.05095 [cs].

[Cai et al., 2021] Cai, R., Zhang, H., Liu, W., Gao, S., and Hao, Z. (2021). Appearance-motion memory consistency network for video anomaly detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 938–946. AAAI Press.

[Caron et al., 2018] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

[Caron et al., 2020] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924.

[Carreira and Zisserman, 2017a] Carreira, J. and Zisserman, A. (2017a). Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

[Carreira and Zisserman, 2017b] Carreira, J. and Zisserman, A. (2017b). Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Catellano, 2014] Catellano, B. (2014). Pyscenedetect: an intelligent scene cut detection and video splitting tool. `https://github.com/Breakthrough/PySceneDetect`.

[Chang et al., 2020] Chang, Y., Zhigang, T., Wei, X., and Junsong, Y. (2020). Clustering driven deep autoencoder for video anomaly detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[Chao et al., 2018] Chao, Y.-W., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., and Sukthankar, R. (2018). Rethinking the faster R-CNN architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Chen et al., 2023] Chen, H., Zhang, W., Wang, Y., and Yang, X. (2023). Improving masked autoencoders by learning where to mask. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 377–390. Springer.

[Chen et al., 2020a] Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. (2020a). Generative pretraining from pixels. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR.

[Chen et al., 2021] Chen, S., Nie, X., Fan, D., Zhang, D., Bhat, V., and Hamid, R. (2021). Shot contrastive self-supervised learning for scene boundary detection.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9796–9805.

[Chen et al., 2020b] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020b). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607. PMLR.

[Chen et al., 2020c] Chen, X., Fan, H., Girshick, R., and He, K. (2020c). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

[Chopra et al., 2005] Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

[Comanici et al., 2025] Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al. (2025). Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

[Crammer and Singer, 2001] Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research - JMLR*, 2.

[Dai et al., 2024] Dai, W., Lee, N., Wang, B., Yang, Z., Liu, Z., Barker, J., Rintamaki, T., Shoeybi, M., Catanzaro, B., and Ping, W. (2024). Nvlm: Open frontier-class multimodal llms. *arXiv preprint arXiv:2409.11402*.

[Dai et al., 2017] Dai, Z., Yang, Z., Yang, F., Cohen, W. W., and Salakhutdinov, R. R. (2017). Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30.

[Damen et al., 2018] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al. (2018).

Scaling egocentric vision: The EPIC-Kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736.

[Dave et al., 2022] Dave, I., Gupta, R., Rizve, M. N., and Shah, M. (2022). TCLR: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406.

[Del Giorno et al., 2016] Del Giorno, A., Bagnell, J. A., and Hebert, M. (2016). A discriminative framework for anomaly detection in large videos. In *European Conference on Computer Vision*, pages 334–349. Springer.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Diba et al., 2019] Diba, A., Sharma, V., Gool, L. V., and Stiefelhagen, R. (2019). Dynamonet: Dynamic action and motion network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6192–6201.

[Ding and Xu, 2018] Ding, L. and Xu, C. (2018). Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6508–6516.

[Dionelis et al., 2020] Dionelis, N., Yaghoobi, M., and Tsaftaris, S. A. (2020). Boundary of distribution support generator (bdsg): Sample generation on the boundary. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 803–807. IEEE.

[Djilali et al., 2021] Djilali, Y. A. D., Krishna, T., McGuinness, K., and O'Connor, N. E. (2021). Rethinking 360deg image visual attention modelling with unsupervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15414–15424.

[Doersch et al., 2015] Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430.

[Dong et al., 2020a] Dong, F., Zhang, Y., and Nie, X. (2020a). Dual discriminator generative adversarial network for video anomaly detection. *IEEE Access*, 8:88170–88176.

[Dong et al., 2020b] Dong, F., Zhang, Y., and Nie, X. (2020b). Dual discriminator generative adversarial network for video anomaly detection. *IEEE Access*, 8:88170–88176.

[Dong and Lin, 2019] Dong, J. and Lin, T. (2019). Margingan: adversarial training in semi-supervised learning. *Advances in neural information processing systems*, 32.

[Dong et al., 2023] Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., Yu, N., and Guo, B. (2023). Peco: Perceptual codebook for bert pre-training of vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 552–560.

[Doshi and Yilmaz, 2020a] Doshi, K. and Yilmaz, Y. (2020a). Any-shot sequential anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 934–935.

[Doshi and Yilmaz, 2020b] Doshi, K. and Yilmaz, Y. (2020b). Continual learning for anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 254–255.

[Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[Du et al., 2022] Du, X., Wang, Z., Cai, M., and Li, Y. (2022). Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*.

[Dumoulin and Visin, 2016] Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

[Esser et al., 2021] Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

[Fabian Caba Heilbron and Niebles, 2015] Fabian Caba Heilbron, Victor Escorcia, B. G. and Niebles, J. C. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970.

[Fan et al., 2023] Fan, D., Wang, J., Liao, S., Zhu, Y., Bhat, V., Santos-Villalobos, H., MV, R., and Li, X. (2023). Motion-guided masking for spatiotemporal representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5619–5629.

[Fan et al., 2021] Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. (2021). Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835.

[Fan et al., 2018] Fan, L., Huang, W., Gan, C., Ermon, S., Gong, B., and Huang, J. (2018). End-to-end learning of motion representation for video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6016–6025.

[Feichtenhofer et al., 2019] Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211.

[Feichtenhofer et al., 2021] Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., and He, K. (2021). A large-scale study on unsupervised spatiotemporal representation

learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3299–3309.

[Feichtenhofer et al., 2022] Feichtenhofer, C., Li, Y., He, K., et al. (2022). Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958.

[Feichtenhofer et al., 2016] Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941.

[Feng and Zhang, 2023] Feng, Z. and Zhang, S. (2023). Evolved part masking for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10386–10395.

[Fernando et al., 2017] Fernando, B., Bilen, H., Gavves, E., and Gould, S. (2017). Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645.

[Fischer et al., 2015] Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., Van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*.

[Gao et al., 2017] Gao, J., Yang, Z., and Nevatia, R. (2017). Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180*.

[Geng et al., 2020] Geng, C., Huang, S.-j., and Chen, S. (2020). Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631.

[Georgescu et al., 2021a] Georgescu, M.-I., Barbalau, A., Ionescu, R. T., Khan, F. S., Popescu, M., and Shah, M. (2021a). Anomaly detection in video via self-

supervised and multi-task learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12742–12752.

[Georgescu et al., 2021b] Georgescu, M. I., Ionescu, R. T., Khan, F. S., Popescu, M., and Shah, M. (2021b). A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4505–4523.

[Gidaris et al., 2018] Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

[Girdhar et al., 2019] Girdhar, R., Carreira, J., Doersch, C., and Zisserman, A. (2019). Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253.

[Girdhar et al., 2023] Girdhar, R., El-Nouby, A., Singh, M., Alwala, K. V., Joulin, A., and Misra, I. (2023). Omnimae: Single model masked pretraining on images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10406–10417.

[Girshick, 2015] Girshick, R. (2015). Fast r-CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1440–1448.

[Goldberg, 2023] Goldberg, Y. (2023). Reinforcement learning for language models. URL `https://gist.github.com/yoavg/6bff0fecd65950898eba1bb321cfbd81`.

[Gong et al., 2019a] Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., and Hengel, A. v. d. (2019a). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[Gong et al., 2019b] Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., and Hengel, A. v. d. (2019b). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1705–1714.

[Goroshin et al., 2015] Goroshin, R., Bruna, J., Tompson, J., Eigen, D., and LeCun, Y. (2015). Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 4086–4093.

[Goyal et al., 2017] Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al. (2017). The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850.

[Grattafiori et al., 2024] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[Gutmann and Hyvärinen, 2010] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

[Han et al., 2017] Han, K., Rezende, R. S., Ham, B., Wong, K.-Y. K., Cho, M., Schmid, C., and Ponce, J. (2017). SCNet: Learning semantic correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 1831–1840.

[Han et al., 2019] Han, T., Xie, W., and Zisserman, A. (2019). Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

[Han et al., 2020a] Han, T., Xie, W., and Zisserman, A. (2020a). Memory-augmented dense predictive coding for video representation learning. In *European conference on computer vision*, pages 312–329. Springer.

[Han et al., 2020b] Han, T., Xie, W., and Zisserman, A. (2020b). Self-supervised co-training for video representation learning. *Advances in neural information processing systems*, 33:5679–5690.

[Hara et al., 2018] Hara, K., Kataoka, H., and Satoh, Y. (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Hasan et al., 2016] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., and Davis, L. S. (2016). Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742.

[He et al., 2022a] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022a). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

[He et al., 2020] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

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

[He et al., 2022b] He, Y., Yang, T., Zhang, Y., Shan, Y., and Chen, Q. (2022b). Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*.

[Hendrycks and Gimpel, 2016] Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

[Hendrycks et al., 2019] Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Hinami et al., 2017] Hinami, R., Mei, T., and Satoh, S. (2017). Joint detection and recounting of abnormal events by learning deep generic knowledge. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3619–3627.

[Hinton and Zemel, 1993] Hinton, G. E. and Zemel, R. S. (1993). Autoencoders, minimum description length and helmholtz free energy. NIPS'93, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Hjelm and Bachman, 2020] Hjelm, R. D. and Bachman, P. (2020). Representation learning with video deep infomax. *arXiv preprint arXiv:2007.13278*.

[Hjelm et al., 2018] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

[Ho et al., 2022] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. (2022). Video diffusion models.

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

[Hong et al., 2021] Hong, D., Li, C., Wen, L., Wang, X., and Zhang, L. (2021). Generic event boundary detection challenge at CVPR 2021 technical report: Cascaded temporal attention network (CASTANET). *arXiv preprint arXiv:2107.00239*.

[Howard et al., 2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[Huang et al., 2023] Huang, B., Zhao, Z., Zhang, G., Qiao, Y., and Wang, L. (2023). Mgmae: Motion guided masking for video masked autoencoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13493–13504.

[Huang et al., 2016] Huang, D.-A., Fei-Fei, L., and Niebles, J. C. (2016). Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153. Springer.

[Hwang et al., 2024] Hwang, S., Yoon, J., Lee, Y., and Hwang, S. J. (2024). Everest: Efficient masked video autoencoder by removing redundant spatiotemporal tokens. In *International Conference on Machine Learning*.

[Ilg et al., 2017] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470.

[Ioffe, 2015] Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[Ionescu et al., 2019a] Ionescu, R. T., Khan, F. S., Georgescu, M.-I., and Shao, L. (2019a). Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7842–7851.

[Ionescu et al., 2019b] Ionescu, R. T., Smeureanu, S., Popescu, M., and Alexe, B. (2019b). Detecting abnormal events in video using narrowed normality clusters. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1951–1960. IEEE.

[Isola et al., 2017] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

[Jang et al., 2016] Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

[Jenni et al., 2020] Jenni, S., Meishvili, G., and Favaro, P. (2020). Video representation learning by recognizing temporal transformations. In *European Conference on Computer Vision*, pages 425–442. Springer.

[Jewell et al., 2022] Jewell, J. T., Khazaie, V. R., and Mohsenzadeh, Y. (2022). One-class learned encoder-decoder network with adversarial context masking for novelty detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3591–3601.

[Ji et al., 2020] Ji, X., Li, B., and Zhu, Y. (2020). Tam-net: Temporal enhanced appearance-to-motion generative network for video anomaly detection. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[Jiang et al., 2019] Jiang, B., Wang, M., Gan, W., Wu, W., and Yan, J. (2019). STM: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2000–2009.

[Jing et al., 2018] Jing, L., Yang, X., Liu, J., and Tian, Y. (2018). Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*.

[Jolliffe, 2005] Jolliffe, I. T. (2005). Principal component analysis and factor analysis. In *Principal Component Analysis*, Springer Series in Statistics. Springer.

[Jordan, 1990] Jordan, M. I. (1990). *Attractor dynamics and parallelism in a connectionist sequential machine*, page 112–127. IEEE Press.

[Kakogeorgiou et al., 2022] Kakogeorgiou, I., Gidaris, S., Psomas, B., Avrithis, Y., Bursuc, A., Karantzalos, K., and Komodakis, N. (2022). What to hide from your students: Attention-guided masked image modeling. In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T., editors, *Computer Vision – ECCV 2022*, pages 300–318, Cham. Springer Nature Switzerland.

[Kang et al., 2021a] Kang, H., Kim, J., Kim, K., Kim, T., and Kim, S. J. (2021a). Winning the CVPR'2021 kinetics-gebd challenge: Contrastive learning approach. *arXiv preprint arXiv:2106.11549*.

[Kang et al., 2021b] Kang, H., Kim, J., Kim, T., and Kim, S. J. (2021b). UBoCo: Unsupervised boundary contrastive learning for generic event boundary detection. *arXiv preprint arXiv:2111.14799*.

[Kay et al., 2017] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

[Kenton and Toutanova, 2019] Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

[Kim et al., 2019] Kim, D., Cho, D., and Kweon, I. S. (2019). Self-supervised video representation learning with space-time cubic puzzles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8545–8552.

[Kingma, 2014] Kingma, D. P. (2014). Auto-encoding variational bayes. *ICLR*.

[Knights et al., 2021] Knights, J., Harwood, B., Ward, D., Vanderkop, A., Mackenzie-Ross, O., and Moghadam, P. (2021). Temporally coherent embeddings for self-supervised video representation learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8914–8921. IEEE.

[Kong et al., 2020] Kong, Q., Wei, W., Deng, Z., Yoshinaga, T., and Murakami, T. (2020). Cycle-contrast for self-supervised video representation learning. *Advances in Neural Information Processing Systems*, 33:8089–8100.

[Krishna et al., 2021] Krishna, T., McGuinness, K., and O'Connor, N. (2021). Evaluating contrastive models for instance-based image retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 471–475.

[Krizhevsky et al., 2012a] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). 2012 AlexNet. *Advances in Neural Information Processing Systems*, pages 1–9.

[Krizhevsky et al., 2012b] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105.

[Kuang et al., 2021] Kuang, H., Zhu, Y., Zhang, Z., Li, X., Tighe, J., Schwertfeger, S., Stachniss, C., and Li, M. (2021). Video contrastive learning with global context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3195–3204.

[Kuehne et al., 2014] Kuehne, H., Arslan, A., and Serre, T. (2014). The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787.

[Kuehne et al., 2011] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE.

[Kwon et al., 2020] Kwon, H., Kim, M., Kwak, S., and Cho, M. (2020). Motionsqueeze: Neural motion feature learning for video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 345–362. Springer.

[Le Cun et al., 1997] Le Cun, Y., Bottou, L., and Bengio, Y. (1997). Reading checks with multilayer graph transformer networks. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 151–154. IEEE.

[Lea et al., 2016a] Lea, C., Reiter, A., Vidal, R., and Hager, G. D. (2016a). Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 36–52, Cham. Springer International Publishing.

[Lea et al., 2016b] Lea, C., Reiter, A., Vidal, R., and Hager, G. D. (2016b). Segmental spatiotemporal CNNs for fine-grained action segmentation. In *European conference on computer vision*, pages 36–52. Springer.

[LeCun, 1998] LeCun, Y. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, pages 2278–2324.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551.

[Lee et al., 2017] Lee, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2017). Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE international conference on computer vision*, pages 667–676.

[Lee et al., 2019a] Lee, J., Kim, D., Ponce, J., and Ham, B. (2019a). SFNet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2278–2287.

[Lee et al., 2020a] Lee, J.-H., Zaheer, M. Z., Astrid, M., and Lee, S.-I. (2020a). Smoothmix: A simple yet effective data augmentation to train robust classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[Lee et al., 2018a] Lee, M., Lee, S., Son, S., Park, G., and Kwak, N. (2018a). Motion feature network: Fixed motion filter for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–403.

[Lee et al., 2018b] Lee, S., Kim, H. G., and Ro, Y. M. (2018b). Stan: Spatio-temporal adversarial networks for abnormal event detection. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1323–1327. IEEE.

[Lee et al., 2019b] Lee, S., Kim, H. G., and Ro, Y. M. (2019b). Bman: Bidirectional multi-scale aggregation networks for abnormal event detection. *IEEE Transactions on Image Processing*, 29:2395–2408.

[Lee et al., 2020b] Lee, S., Kim, H. G., and Ro, Y. M. (2020b). Bman: Bidirectional multi-scale aggregation networks for abnormal event detection. *IEEE Transactions on Image Processing*, 29:2395–2408.

[Li et al., 2022a] Li, C., Wang, X., Hong, D., Wang, Y., Zhang, L., Luo, T., and Wen, L. (2022a). Structured context transformer for generic event boundary detection. *arXiv preprint arXiv:2206.02985*.

[Li et al., 2022b] Li, C., Wang, X., Wen, L., Hong, D., Luo, T., and Zhang, L. (2022b). End-to-end compressed video representation learning for generic event boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13967–13976.

[Li et al., 2021a] Li, F., Li, G., He, X., and Cheng, J. (2021a). Dynamic dual gating neural networks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5310–5319.

[Li et al., 2022c] Li, G., Zheng, H., Liu, D., Wang, C., Su, B., and Zheng, C. (2022c). Semmae: Semantic-guided masking for learning masked autoencoders. *Advances in Neural Information Processing Systems*, 35:14290–14302.

[Li et al., 2021b] Li, R., Zhang, Y., Qiu, Z., Yao, T., Liu, D., and Mei, T. (2021b). Motion-focused contrastive learning of video representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2105–2114.

[Li et al., 2014] Li, W., Mahadevan, V., and Vasconcelos, N. (2014). Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32.

[Li et al., 2022d] Li, X., Wang, W., Yang, L., and Yang, J. (2022d). Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality. *arXiv preprint arXiv:2205.10063*.

[Liang et al., 2017] Liang, X., Lee, L., Dai, W., and Xing, E. P. (2017). Dual motion gan for future-flow embedded video prediction. In *proceedings of the IEEE international conference on computer vision*, pages 1744–1752.

[Likas et al., 2003] Likas, A., Vlassis, N., and J. Verbeek, J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461. Biometrics.

[Lin et al., 2019a] Lin, J., Gan, C., and Han, S. (2019a). TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7083–7093.

[Lin et al., 2019b] Lin, T., Liu, X., Li, X., Ding, E., and Wen, S. (2019b). BMN: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898.

[Lin et al., 2018] Lin, T., Zhao, X., Su, H., Wang, C., and Yang, M. (2018). BSN: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19.

[Lin et al., 2022] Lin, X., Chen, Y., Li, G., and Yu, Y. (2022). A causal inference look at unsupervised video anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1620–1629.

[Liu et al., 2018a] Liu, W., Luo, W., Lian, D., and Gao, S. (2018a). Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6536–6545.

[Liu et al., 2019] Liu, X., Lee, J.-Y., and Jin, H. (2019). Learning video representations from correspondence proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4281.

[Liu et al., 2018b] Liu, Y., Li, C.-L., and Póczos, B. (2018b). Classifier two sample test for video anomaly detections. In *BMVC*, page 71.

[Liu et al., 2021] Liu, Z., Nie, Y., Long, C., Zhang, Q., and Li, G. (2021). A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13588–13597.

[Lorre et al., 2020] Lorre, G., Rabarisoa, J., Orcesi, A., Ainouz, S., and Canu, S. (2020). Temporal contrastive pretraining for video action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 662–670.

[Lotter et al., 2017] Lotter, W., Kreiman, G., and Cox, D. (2017). Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations*.

[Lu et al., 2013] Lu, C., Shi, J., and Jia, J. (2013). Abnormal event detection at 150 fps in matlab. In *2013 IEEE International Conference on Computer Vision*, pages 2720–2727.

[Lu et al., 2019] Lu, Y., Kumar, K. M., shahabeddin Nabavi, S., and Wang, Y. (2019). Future frame prediction using convolutional vrnn for anomaly detection.

In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE.

[Lu et al., 2020] Lu, Y., Yu, F., Reddy, M. K. K., and Wang, Y. (2020). Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*, pages 125–141. Springer.

[Luo et al., 2017a] Luo, W., Liu, W., and Gao, S. (2017a). Remembering history with convolutional lstm for anomaly detection. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 439–444.

[Luo et al., 2017b] Luo, W., Liu, W., and Gao, S. (2017b). Remembering history with convolutional lstm for anomaly detection. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 439–444. IEEE.

[Luo et al., 2017c] Luo, W., Liu, W., and Gao, S. (2017c). A revisit of sparse coding based anomaly detection in stacked rnn framework. *ICCV, Oct*, 1(2):3.

[Luo et al., 2017d] Luo, W., Liu, W., and Gao, S. (2017d). A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[Maaz et al., 2024] Maaz, M., Rasheed, H., Khan, S., and Khan, F. (2024). Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602.

[Madan et al., 2024] Madan, N., Ristea, N.-C., Nasrollahi, K., Moeslund, T. B., and Ionescu, R. T. (2024). Cl-mae: Curriculum-learned masked autoencoders. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2492–2502.

[Mahadevan et al., 2010] Mahadevan, V., Li, W., Bhalodia, V., and Vasconcelos, N. (2010). Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981. IEEE.

[Materzynska et al., 2019] Materzynska, J., Berger, G., Bax, I., and Memisevic, R. (2019). The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0.

[Mathieu et al., 2016] Mathieu, M., Couprie, C., and LeCun, Y. (2016). Deep multi-scale video prediction beyond mean square error. In *4th International Conference on Learning Representations, ICLR 2016*.

[McLachlan and Krishnan, 2008] McLachlan, G. J. and Krishnan, T. (2008). *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics. Wiley.

[Meng et al., 2020] Meng, Y., Panda, R., Lin, C.-C., Sattigeri, P., Karlinsky, L., Saenko, K., Oliva, A., and Feris, R. (2020). Adafuse: Adaptive temporal fusion network for efficient action recognition. In *International Conference on Learning Representations*.

[Miech et al., 2019a] Miech, A., Laptev, I., Sivic, J., Wang, H., Torresani, L., and Tran, D. (2019a). Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[Miech et al., 2019b] Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. (2019b). Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640.

[Min et al., 2019] Min, J., Lee, J., Ponce, J., and Cho, M. (2019). Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3395–3404.

[Mirzaei et al., 2023] Mirzaei, H., Salehi, M., Shahabi, S., Gavves, E., Snoek, C. G. M., Sabokrou, M., and Rohban, M. H. (2023). Fake it until you make it : To-

wards accurate near-distribution novelty detection. In *The Eleventh International Conference on Learning Representations*.

[Misra and Maaten, 2020] Misra, I. and Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717.

[Misra et al., 2016] Misra, I., Zitnick, C. L., and Hebert, M. (2016). Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer.

[Mnih and Kavukcuoglu, 2013] Mnih, A. and Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. *Advances in neural information processing systems*, 26.

[Neimark et al., 2021] Neimark, D., Bar, O., Zohar, M., and Asselmann, D. (2021). Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3163–3172.

[Ng et al., 2011] Ng, A. et al. (2011). Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19.

[Ngo et al., 2019] Ngo, P. C., Winarto, A. A., Kou, C. K. L., Park, S., Akram, F., and Lee, H. K. (2019). Fence gan: Towards better anomaly detection. In *2019 IEEE 31St International Conference on tools with artificial intelligence (ICTAI)*, pages 141–148. IEEE.

[Nguyen et al., 2024] Nguyen, D. K., Li, Y., Aggarwal, V., Oswald, M. R., Kirillov, A., Snoek, C. G. M., and Chen, X. (2024). R-MAE: Regions meet masked autoencoders. In *The Twelfth International Conference on Learning Representations*.

[Nguyen and Meunier, 2019] Nguyen, T.-N. and Meunier, J. (2019). Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1273–1283.

[Noroozi and Favaro, 2016] Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer.

[Novotny et al., 2018] Novotny, D., Larlus, D., and Vedaldi, A. (2018). Capturing the geometry of object categories from video supervision. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):261–275.

[Oord et al., 2018] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[Oquab et al., 2023] Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision.

[Pan et al., 2021] Pan, T., Song, Y., Yang, T., Jiang, W., and Liu, W. (2021). Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11205–11214.

[Pang et al., 2020] Pang, G., Yan, C., Shen, C., Hengel, A. v. d., and Bai, X. (2020). Self-trained deep ordinal regression for end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12173–12182.

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

[Pathak et al., 2017] Pathak, D., Girshick, R., Dollár, P., Darrell, T., and Hariharan, B. (2017). Learning features by watching objects move. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2701–2710.

[Patrick et al., 2021] Patrick, M., Campbell, D., Asano, Y., Misra, I., Metze, F., Feichtenhofer, C., Vedaldi, A., and Henriques, J. F. (2021). Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506.

[Patrick et al., 2020] Patrick, M., Huang, P.-Y., Asano, Y., Metze, F., Hauptmann, A., Henriques, J., and Vedaldi, A. (2020). Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*.

[Piergiovanni and Ryoo, 2019] Piergiovanni, A. and Ryoo, M. S. (2019). Representation flow for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9945–9953.

[Pirsiavash and Ramanan, 2014] Pirsiavash, H. and Ramanan, D. (2014). Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Pourreza et al., 2021] Pourreza, M., Mohammadi, B., Khaki, M., Bouindour, S., Snoussi, H., and Sabokrou, M. (2021). G2d: Generate to detect anomaly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2003–2012.

[Qian et al., 2021a] Qian, R., Li, Y., Yuan, L., Gong, B., Liu, T., Brown, M., Belongie, S., Yang, M.-H., Adam, H., and Cui, Y. (2021a). Exploring temporal granularity in self-supervised video representation learning. *arXiv preprint arXiv:2112.04480*.

[Qian et al., 2021b] Qian, R., Meng, T., Gong, B., Yang, M.-H., Wang, H., Belongie, S., and Cui, Y. (2021b). Spatiotemporal contrastive video representation

learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6964–6974.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

[Rai et al., 2021a] Rai, A. K., Krishna, T., Dietlmeier, J., McGuinness, K., Smeaton, A. F., and O'Connor, N. E. (2021a). Discerning generic event boundaries in long-form wild videos. *arXiv preprint arXiv:2106.10090*.

[Rai et al., 2021b] Rai, N., Adeli, E., Lee, K.-H., Gaidon, A., and Niebles, J. C. (2021b). Cocon: Cooperative-contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3384–3393.

[Ramachandra and Jones, 2020] Ramachandra, B. and Jones, M. (2020). Street scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2569–2578.

[Ramachandra et al., 2020a] Ramachandra, B., Jones, M., and Vatsavai, R. (2020a). Learning a distance function with a siamese network to localize anomalies in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2598–2607.

[Ramachandra et al., 2020b] Ramachandra, B., Jones, M. J., and Vatsavai, R. R. (2020b). A survey of single-scene video anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2293–2312.

[Ramesh et al., 2021] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.

[Rasheed et al., 2023] Rasheed, H., Khattak, M. U., Maaz, M., Khan, S., and Khan, F. S. (2023). Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6545–6554.

[Ravanbakhsh et al., 2017] Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., and Sebe, N. (2017). Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1577–1581. IEEE.

[Ren et al., 2015a] Ren, S., He, K., Girshick, R., and Sun, J. (2015a). Faster r-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28.

[Ren et al., 2015b] Ren, S., He, K., Girshick, R., and Sun, J. (2015b). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

[Reynolds, 2009] Reynolds, D. A. (2009). Gaussian mixture models. In *Encyclopedia of Biometrics*. Springer.

[Rifai et al., 2011] Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., Dauphin, Y., and Glorot, X. (2011). Higher order contractive auto-encoder. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*, pages 645–660. Springer.

[Rocco et al., 2017] Rocco, I., Arandjelovic, R., and Sivic, J. (2017). Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157.

[Rombach et al., 2022] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

[Rosenblatt, 1957] Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

[Santoro et al., 2017] Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. (2017). A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30.

[Schiappa et al., 2022] Schiappa, M. C., Rawat, Y. S., and Shah, M. (2022). Self-supervised learning for videos: A survey. *ArXiv*, abs/2207.00419.

[Schiappa et al., 2023] Schiappa, M. C., Rawat, Y. S., and Shah, M. (2023). Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 55(13s):1–37.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[Shao et al., 2020a] Shao, D., Zhao, Y., Dai, B., and Lin, D. (2020a). Intra- and inter-action understanding via temporal action parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Shao et al., 2020b] Shao, D., Zhao, Y., Dai, B., and Lin, D. (2020b). Intra- and inter-action understanding via temporal action parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 727–736.

[Shi et al., 2022] Shi, Y., Siddharth, N., Torr, P., and Kosiorek, A. R. (2022). Adversarial masking for self-supervised learning. In *International Conference on Machine Learning*, pages 20026–20040. PMLR.

[Shou et al., 2021] Shou, M. Z., Lei, S. W., Wang, W., Ghadiyaram, D., and Feiszli, M. (2021). Generic event boundary detection: A benchmark for event segmentation. *arXiv preprint arXiv:2101.10511*.

[Sigurdsson et al., 2016] Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., and Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems*, 27.

[Singh et al., 2023] Singh, A., Jones, M. J., and Learned-Miller, E. G. (2023). Eval: Explainable video anomaly localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18717–18726.

[Smeureanu et al., 2017] Smeureanu, S., Ionescu, R. T., Popescu, M., and Alexe, B. (2017). Deep appearance features for abnormal behavior detection in video. In *International Conference on Image Analysis and Processing*, pages 779–789. Springer.

[Sohl-Dickstein et al., 2015] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

[Song et al., 2020] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

[Soomro et al., 2012] Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

[Sultani et al., 2018] Sultani, W., Chen, C., and Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488.

[Sun et al., 2019] Sun, C., Baradel, F., Murphy, K., and Schmid, C. (2019). Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*.

[Sun et al., 2020] Sun, C., Jia, Y., Hu, Y., and Wu, Y. (2020). Scene-aware context reasoning for unsupervised abnormal event detection in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 184–192.

[Sun et al., 2018a] Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018a). Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943.

[Sun et al., 2018b] Sun, S., Kuang, Z., Sheng, L., Ouyang, W., and Zhang, W. (2018b). Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1390–1399.

[Sun et al., 2023] Sun, X., Chen, P., Chen, L., Li, C., Li, T. H., Tan, M., and Gan, C. (2023). Masked motion encoding for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2235–2245.

[Suvorov et al., 2022] Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., and Lempitsky, V. (2022). Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159.

[Tang et al., 2021] Tang, J., Liu, Z., Qian, C., Wu, W., and Wang, L. (2021). Progressive attention on multi-level dense difference maps for generic event boundary detection. *arXiv preprint arXiv:2112.04771*.

[Tang et al., 2020a] Tang, Y., Zhao, L., Zhang, S., Gong, C., Li, G., and Yang, J. (2020a). Integrating prediction and reconstruction for anomaly detection. *Pattern Recognition Letters*, 129:123–130.

[Tang et al., 2020b] Tang, Z., Gao, Y., Karlinsky, L., Sattigeri, P., Feris, R., and Metaxas, D. (2020b). Onlineaugment: Online data augmentation with less domain knowledge. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 313–329. Springer.

[Tao et al., 2020] Tao, L., Wang, X., and Yamasaki, T. (2020). Self-supervised video representation learning using inter-intra contrastive framework. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2193–2201.

[Team et al., 2024] Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

[Teed and Deng, 2020] Teed, Z. and Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer.

[Tian et al., 2020a] Tian, Y., Che, Z., Bao, W., Zhai, G., and Gao, Z. (2020a). Self-supervised motion representation via scattering local motion cues. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV*, page 71–89. Springer-Verlag.

[Tian et al., 2020b] Tian, Y., Krishnan, D., and Isola, P. (2020b). Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer.

[Tong et al., 2022] Tong, Z., Song, Y., Wang, J., and Wang, L. (2022). VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*.

[Touvron et al., 2021] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR.

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

[Tran et al., 2015] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.

[Tran et al., 2018] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

[Tulyakov et al., 2018] Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018). Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535.

[Tur et al., 2023a] Tur, A. O., Dall'Asen, N., Beyan, C., and Ricci, E. (2023a). Exploring diffusion models for unsupervised video anomaly detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2540–2544.

[Tur et al., 2023b] Tur, A. O., Dall'Asen, N., Beyan, C., and Ricci, E. (2023b). Unsupervised video anomaly detection with diffusion models conditioned on compact motion representations. In *International Conference on Image Analysis and Processing*, pages 49–62. Springer.

[Tversky and Zacks, 2013] Tversky, B. and Zacks, J. M. (2013). Event perception. *Oxford Handbook of Cognitive Psychology*, 1(2):3.

[Van Den Oord et al., 2017] Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

[Varol et al., 2017] Varol, G., Laptev, I., and Schmid, C. (2017). Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[Veit and Belongie, 2018] Veit, A. and Belongie, S. (2018). Convolutional networks with adaptive inference graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18.

[Vincent et al., 2008a] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008a). Extracting and composing robust features with denoising autoencoders.

In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

[Vincent et al., 2008b] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008b). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA. Association for Computing Machinery.

[Vincent et al., 2010] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12).

[Voleti et al., 2022] Voleti, V., Jolicoeur-Martineau, A., and Pal, C. (2022). Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in Neural Information Processing Systems*, 35:23371–23385.

[von Luxburg, 2007] von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

[Vondrick et al., 2016] Vondrick, C., Pirsiavash, H., and Torralba, A. (2016). Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106.

[Vondrick et al., 2018] Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., and Murphy, K. (2018). Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408.

[Vu et al., 2019] Vu, H., Nguyen, T. D., Le, T., Luo, W., and Phung, D. (2019). Robust anomaly detection in videos using multilevel representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5216–5223.

[Wang et al., 2022a] Wang, G., Wang, Y., Qin, J., Zhang, D., Bao, X., and Huang, D. (2022a). Video anomaly detection by solving decoupled spatio-temporal jigsaw

puzzles. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 494–511. Springer.

[Wang et al., 2019a] Wang, J., Jiao, J., Bao, L., He, S., Liu, Y., and Liu, W. (2019a). Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4006–4015.

[Wang et al., 2019b] Wang, J., Jiao, J., Bao, L., He, S., Liu, Y., and Liu, W. (2019b). Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Wang et al., 2020a] Wang, J., Jiao, J., and Liu, Y.-H. (2020a). Self-supervised video representation learning by pace prediction. In *European conference on computer vision*, pages 504–521. Springer.

[Wang et al., 2023a] Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., and Qiao, Y. (2023a). Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560.

[Wang et al., 2016] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer.

[Wang et al., 2022b] Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Jiang, Y.-G., Zhou, L., and Yuan, L. (2022b). Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14733–14743.

[Wang et al., 2023b] Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Yuan, L., and Jiang, Y.-G. (2023b). Masked video distillation: Rethinking masked

feature modeling for self-supervised video representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6312–6322.

[Wang and Gupta, 2015] Wang, X. and Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802.

[Wang et al., 2019c] Wang, X., Jabri, A., and Efros, A. A. (2019c). Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2566–2576.

[Wang et al., 2022c] Wang, Y., Gao, D., Yu, L., Lei, S. W., Feiszli, M., and Shou, M. Z. (2022c). Generic event boundary captioning: A benchmark for status changes understanding. *arXiv preprint arXiv:2204.00486*.

[Wang et al., 2022d] Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al. (2022d). Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*.

[Wang et al., 2020b] Wang, Z., Zou, Y., and Zhang, Z. (2020b). Cluster attention contrast for video anomaly detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2463–2471.

[Wei et al., 2022] Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. (2022). Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678.

[Wei et al., 2018] Wei, D., Lim, J. J., Zisserman, A., and Freeman, W. T. (2018). Learning and using the arrow of time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8052–8060.

[Wei et al., 2024] Wei, Z., Pan, Z., and Owens, A. (2024). Efficient vision-language pre-training by cluster masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26815–26825.

[Williams, 1992] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

[Wiskott and Sejnowski, 2002] Wiskott, L. and Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770.

[Wolleb et al., 2022] Wolleb, J., Bieder, F., Sandkühler, R., and Cattin, P. C. (2022). Diffusion models for medical anomaly detection. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*, pages 35–45. Springer.

[Wu et al., 2019] Wu, P., Liu, J., and Shen, F. (2019). A deep one-class neural network for anomalous event detection in complex scenes. *IEEE transactions on neural networks and learning systems*, 31(7):2609–2622.

[Wu et al., 2020] Wu, P., Liu, J., Shi, Y., Sun, Y., Shao, F., Wu, Z., and Yang, Z. (2020). Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 322–339. Springer.

[Wu et al., 2018] Wu, Z., Xiong, Y., Yu, S., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*.

[Xia and Zhan, 2020] Xia, H. and Zhan, Y. (2020). A survey on temporal action localization. *IEEE Access*, 8:70477–70487.

[Xie et al., 2023] Xie, J., Li, W., Zhan, X., Liu, Z., Ong, Y.-S., and Loy, C. C. (2023). Masked frequency modeling for self-supervised visual pre-training. In *The Eleventh International Conference on Learning Representations*.

[Xie et al., 2017] Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

[Xie et al., 2018] Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321.

[Xie et al., 2022] Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. (2022). Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663.

[Xu et al., 2015] Xu, D., Ricci, E., Yan, Y., Song, J., and Sebe, N. (2015). Learning deep representations of appearance and motion for anomalous event detection. In *BMVC*.

[Xu et al., 2014] Xu, D., Song, R., Wu, X., Li, N., Feng, W., and Qian, H. (2014). Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. *Neurocomputing*, 143:144–152.

[Xu et al., 2019] Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., and Zhuang, Y. (2019). Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343.

[Xu et al., 2017] Xu, D., Yan, Y., Ricci, E., and Sebe, N. (2017). Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117–127.

[Xu et al., 2021] Xu, H., Ghosh, G., Huang, P.-Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L., and Feichtenhofer, C. (2021). Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*.

[Yan et al., 2023] Yan, C., Zhang, S., Liu, Y., Pang, G., and Wang, W. (2023). Feature prediction diffusion model for video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5527–5537.

[Yan et al., 2022] Yan, S., Zhu, T., Wang, Z., Cao, Y., Zhang, M., Ghosh, S., Wu, Y., and Yu, J. (2022). Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*.

[Yang et al., 2020a] Yang, C., Xu, Y., Dai, B., and Zhou, B. (2020a). Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*.

[Yang et al., 2022a] Yang, H., Huang, D., Wen, B., Wu, J., Yao, H., Jiang, Y., Zhu, X., and Yuan, Z. (2022a). Self-supervised video representation learning with motion-aware masked autoencoders. *arXiv preprint arXiv:2210.04154*.

[Yang et al., 2020b] Yang, X., Yang, X., Liu, S., Sun, D., Davis, L., and Kautz, J. (2020b). Hierarchical contrastive motion learning for video action recognition. *arXiv preprint arXiv:2007.10321*.

[Yang et al., 2023] Yang, Z., Liu, J., Wu, Z., Wu, P., and Liu, X. (2023). Video event restoration based on keyframes for video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14592–14601.

[Yang et al., 2022b] Yang, Z., Wu, P., Liu, J., and Liu, X. (2022b). Dynamic local aggregation network with adaptive clusterer for anomaly detection. In *European Conference on Computer Vision*, pages 404–421. Springer.

[Yao et al., 2020] Yao, Y., Liu, C., Luo, D., Zhou, Y., and Ye, Q. (2020). Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6548–6557.

[Ye et al., 2019a] Ye, M., Peng, X., Gan, W., Wu, W., and Qiao, Y. (2019a). Anopcn: Video anomaly detection via deep predictive coding network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1805–1813.

[Ye et al., 2019b] Ye, M., Zhang, X., Yuen, P. C., and Chang, S.-F. (2019b). Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219.

[Yu et al., 2020] Yu, G., Wang, S., Cai, Z., Zhu, E., Xu, C., Yin, J., and Kloft, M. (2020). Cloze test helps: Effective video anomaly detection via learning to complete video events. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 583–591.

[Zach et al., 2007] Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007. Proceedings 29*, pages 214–223. Springer.

[Zaheer et al., 2020a] Zaheer, M. Z., Lee, J.-h., Astrid, M., and Lee, S.-I. (2020a). Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193.

[Zaheer et al., 2020b] Zaheer, M. Z., Lee, J.-H., Astrid, M., Mahmood, A., and Lee, S.-I. (June 2020b). Cleaning label noise with clusters for minimally supervised anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

[Zaheer et al., 2022a] Zaheer, M. Z., Lee, J.-H., Mahmood, A., Astrid, M., and Lee, S.-I. (2022a). Stabilizing adversarially learned one-class novelty detection using pseudo anomalies. *IEEE Transactions on Image Processing*, 31:5963–5975.

[Zaheer et al., 2020c] Zaheer, M. Z., Mahmood, A., Astrid, M., and Lee, S.-I. (2020c). Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In *European Conference on Computer Vision*, pages 358–376. Springer.

[Zaheer et al., 2022b] Zaheer, M. Z., Mahmood, A., Khan, M. H., Segu, M., Yu, F., and Lee, S.-I. (2022b). Generative cooperative learning for unsupervised video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14744–14754.

[Zaheer et al., 2020d] Zaheer, M. Z., Mahmood, A., Shin, H., and Lee, S.-I. (2020d). A self-reasoning framework for anomaly detection using video-level labels. *IEEE Signal Processing Letters*, 27:1705–1709.

[Zbontar et al., 2021] Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, pages 12310–12320. PMLR.

[Zhang et al., 2023a] Zhang, C., Li, G., Qi, Y., Wang, S., Qing, L., Huang, Q., and Yang, M.-H. (2023a). Exploiting completeness and uncertainty of pseudo labels for weakly supervised video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16271–16280.

[Zhang et al., 2018] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

[Zhang et al., 2023b] Zhang, H., Li, X., and Bing, L. (2023b). Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

[Zhang et al., 2022] Zhang, Q., Wang, Y., and Wang, Y. (2022). How mask matters: Towards theoretical understandings of masked autoencoders. *Advances in Neural Information Processing Systems*, 35:27127–27139.

[Zhang et al., 2019] Zhang, X., Wang, Q., Zhang, J., and Zhong, Z. (2019). Adversarial autoaugment. *arXiv preprint arXiv:1912.11188*.

[Zhang et al., 2016] Zhang, Y., Lu, H., Zhang, L., Ruan, X., and Sakai, S. (2016). Video anomaly detection based on locality sensitive hashing filters. *Pattern Recognition*, 59:302–311.

[Zhao and Dong, 2020] Zhao, Q. and Dong, J. (2020). Self-supervised representation learning by predicting visual permutations. *Knowledge-Based Systems*, 210:106534.

[Zhao et al., 2017a] Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., and Hua, X.-S. (2017a). 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. Association for Computing Machinery.

[Zhao et al., 2017b] Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., and Hua, X.-S. (2017b). Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941.

[Zhao et al., 2018] Zhao, Y., Xiong, Y., and Lin, D. (2018). Recognize actions by disentangling components of dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6566–6575.

[Zhou et al., 2018] Zhou, B., Andonian, A., Oliva, A., and Torralba, A. (2018). Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 803–818.

[Zhou et al., 2017] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Zhou et al., 2022] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. (2022). Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*.

[Zhu et al., 2022] Zhu, Y., Bao, W., and Yu, Q. (2022). Towards open set video anomaly detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 395–412. Springer.