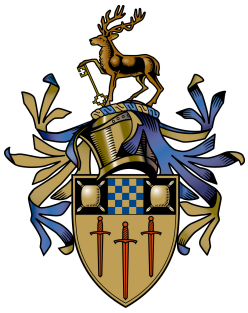
# Advancing Efficiency and Accessibility in Multimodal Video Understanding with Deep Learning

Edward Fish

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey



Faculty of Arts and Social Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

June 2024

# Abstract

In the rapidly expanding digital landscape, the ability to extract meaningful insights from vast quantities of video content is transformative. However, many organisations face a critical challenge: they lack the substantial computational resources and the time-intensive annotation processes required to leverage advanced video analysis technologies fully. This thesis addresses this gap by introducing several resource-efficient deep learning strategies tailored for multimodal video understanding applications. The presented methodologies focus on leveraging pre-trained foundational neural networks for multimodal feature extraction, fusion, and spatiotemporal understanding.

First, we present a method for fusing multimodal features from video for enhanced style and semantic clustering with weakly labelled video data. By tapping into the capabilities of pre-trained foundational models, we develop a method that captures intricate contextual cues within multimodal video data for improved semantic video recommendation and retrieval applications.

Advancing the challenge of long video understanding, we present an innovative architecture that utilises pre-trained encoders to extract spatiotemporal features at various resolutions. This approach achieves state-of-the-art performance in tasks requiring fine-grained temporal analysis, such as speaker recognition and character identification while maintaining computational efficiency.

We continue with a novel approach to audio-visual fusion for temporal action localisation by introducing a gated cross-attention mechanism, which effectively integrates audio and visual features for activity recognition and localisation applications. This results in a low-parameter solution that optimises data utility while improving performance over uni-modal approaches.

The final contribution of this thesis is developing a technique for aligning text prompts with visual features using prompt learning and optimal transport. This strategy significantly reduces training overhead and improves generalisation by leveraging pre-trained visual-language features and optimising only a few learnable parameters. This enables precise action localisation and discrimination between foreground and background elements using only a few labelled samples per class.

Collectively, these contributions improve access to advanced video analytical tools, making them more accessible to a broader audience, including those constrained by computational and financial limitations. This work advances the technical boundaries of video analysis and democratises its applications, fostering innovation across various fields.

**Key words:** Deep Learning, Machine Learning, Video Understanding, Action Recognition, Action Localization

Email: ef0036@surrey.ac.uk
WWW: https://ed-fish.github.io/

# Acknowledgements

This thesis would not have been possible without the invaluable support of my supervisors, friends, and family.

First, I would like to express my gratitude to my supervisor, Jon Weinbren. Thank you for providing me with the opportunity to pursue this research, helping to secure funding, and fostering a creative, supportive, and interdisciplinary environment.

I am also profoundly grateful to my supervisor, Dr. Andrew Gilbert. Your patience, guidance, and dedication have helped me to overcome self-doubt and believe in my potential. Your commitment to your students is truly inspiring, and I feel incredibly fortunate to have worked with you.

To my parents, Terry and Lynne, thank you for instilling in me the importance of education and for your financial support throughout my undergraduate and master's degrees. I hope to create opportunities for those less privileged than myself throughout my academic journey.

Lastly, to my wife, Carli, thank you for your love, patience, and support in every part of my life.

# Declaration

This thesis and the work to which it refers are the results of my own efforts. Any ideas, data, images or text resulting from the work of others (whether published or unpublished) are fully identified as such within the work and attributed to their originator in the text, bibliography or in footnotes. This thesis has not been submitted in whole or in part for any other academic degree or professional qualification. I agree that the University has the right to submit my work to the plagiarism detection service TurnitinUK for originality checks. Whether or not drafts have been so-assessed, the University reserves the right to require an electronic version of the final document (as submitted) for assessment as above.

The work presented in this thesis is also present in the following manuscripts:

1. E. Fish, J. Weinbren, A. Gilbert, "Rethinking Genre Classification with Fine-grained Semantic Clustering," *In Proceedings of IEEE International Conference on Image Processing (ICIP), 2022*. **(Chapter 2)**
2. E. Fish, J. Weinbren, A. Gilbert, "Two-Stream Transformer Architecture for Long Form Video Understanding." *In Proceedings of 33rd British Machine Vision Conference (BMVC), 2022*. **(Chapter 3)**
3. E. Fish, J. Weinbren, A. Gilbert, "Multi-Resolution Audio-Visual Feature Fusion for Temporal Action Localization," *In Proceedings of Neural Information Processing Systems Workshop on Machine Learning for Audio (NeurIPS), 2023*. **(Chapter 4)**
4. E. Fish, J. Weinbren, A. Gilbert, "PLOT-TAL–Prompt Learning with Optimal Transport for Few-Shot Temporal Action Localization," *Under Review*, 2024. **(Chapter 5)**

Signed: Edward Fish

Date: 06/2024

# Contents

# Nomenclature

## Acronyms

| | |
|---|---|
| **HMM** | Hidden Markov Model |
| **DTW** | Dynamic Time Warping |
| **GMM** | Gaussian Mixture Model |
| **EM** | Expectation-Maximization |
| **EKF** | Extended Kalman Filter |
| **UKF** | Unscented Kalman Filter |
| **SVM** | Support Vector Machine |
| **STIP** | Spatiotemporal Interest Point |
| **iDT** | Improved Dense Trajectories |
| **SIFT-3D** | 3D Scale-Invariant Feature Transform |
| **HOG3D** | 3D Histogram of Oriented Gradients |
| **SURF** | Speeded Up Robust Features |
| **CNN** | Convolutional Neural Network |
| **C3D** | 3D Convolutional Network |
| **RNN** | Recurrent Neural Network |
| **LSTM** | Long Short-Term Memory Network |
| **BERT** | Bidirectional Encoder Representations from Transformers |
| **ViViT** | Video Vision Transformer |
| **FPN** | Feature Pyramid Network |
| **LDA** | Latent Dirichlet Allocation |

| | |
|---|---|
| **CCA** | Canonical Correlation Analysis |
| **R(2+1)d** | Spatiotemporal Convolutions Through 3D Convolutional Networks |
| **MLP** | Multi-Layer Perceptron |
| **NT-Xent** | Normalized Temperature-Scaled Cross-Entropy Loss |
| **RGB** | Red Green Blue (Colour Space) |
| **TAL** | Temporal Action Localization |
| **OT** | Optimal Transport |
| **CLIP** | Contrastive Language-Image Pretraining |
| **CoOp** | Context Optimization |
| **CoCoOp** | Conditional Context Optimization |
| **IoU** | Intersection over Union |
| **DIoU** | Distance Intersection over Union |
| **AU(PRC)** | Area Under Precision-Recall Curve |
| **TSNE** | t-Distributed Stochastic Neighbor Embedding |
| **UMAP** | Uniform Manifold Approximation and Projection |
| **ANNOY** | Approximate Nearest Neighbors Oh Yeah |
| **SIMCLR** | Simple Framework for Contrastive Learning of Visual Representations |
| **AMSGRAD** | AMSGrad Optimization Algorithm |
| **TVL1** | Total Variation L1 |
| **FFmpeg** | Fast Forward Moving Picture Experts Group |
| **SE-Net** | Squeeze-and-Excitation Networks |
| **I3D** | Inflated 3D Convolutional Network |
| **LVU** | Long Video Understanding |
| **STAN** | Spatio-Temporal Attention Network |
| **MMX** | MMX-Trailer-20 Dataset |
| **NL** | Non-Local |
| **BCE** | Binary Cross Entropy |
| **VGG** | Visual Geometry Group |
| **mAP** | Mean Average Precision |
| **GELU** | Gaussian Error Linear Units |

| | |
|---|---|
| **CLS** | Classification Token |
| **MRAV-FF** | Multi-Resolution Audio-Visual Feature Fusion |
| **tIoU** | Temporal Intersection over Union |
| **STFT** | Short-Time Fourier Transform |
| **EPIC** | Egocentric Perception Dataset |
| **FC** | Fully Connected |
| **VLP** | Vision-Language Pre-trained |
| **TSP** | Temporal Scale Pooling |
| **THUMOS** | Temporally Annotated Video Corpus |
| **PL** | Prompt Learning |
| **E2E** | End-to-End |
| **ML** | Meta-Learning |
| **GPT** | Generative Pre-Trained Transformer |
| **SOTA** | State Of The Art |
| **GPU** | Graphics Processing Unit |
| **MAMBA** | Linear-Time Sequence Modeling with Selective State Spaces Architecture |

# Symbols

### Introduced in Chapter 3

| | |
|---|---|
| $\sigma$ | Sigmoid function. |
| $\circ$ | Element-wise (Hadamard) product. |
| $\mathbb{V}$ | A set of all frames forming a video |
| $\mathbb{S}$ | A subset of frames forming a set of clips. |
| $\mathbf{c}$ | A sample from clips constituting $T$ number of frames forming a clip. |
| $T$ | Total duration of the video, which is variable. |
| $\mathbb{Y}$ | Ground truth labels. |
| $\{\Psi^1, \Psi^2, \ldots, \Psi^E\}$ | Pre-trained single modality experts. |
| $\Psi^e$ | The $e$'th expert. |
| $\boldsymbol{g}(\cdot)$ | Function used to infer pairwise task relationships. |
| $\boldsymbol{h}(\cdot)$ | Function that maps the sum of all pairwise relationships into a single attention vector. |
| $\boldsymbol{m}(\cdot)$ | Projection head encoder function. |
| $\mathcal{L}_{NTX}(\cdot)$ | Normalised temperature-scaled cross-entropy loss (NT-Xent). |
| $\mathcal{L}_{BCE}(\cdot, \cdot)$ | Binary Cross Entropy Loss with logits. |
| $\mathrm{sim}(\cdot, \cdot)$ | Cosine similarity metric. |
| $\tau$ | Temperature parameter. |
| $\mathbf{x_p}$ | Pair feature representation from the same video. |
| $\mathbf{x_n}$ | Feature representation from another video. |

## Introduced in Chapter 4

| | |
|---|---|
| $\hat{t}$ | Feature embedding token representing the projected temporal features of a scene. |
| $\mathbb{Z}$ | Set of all temporal embedding vectors for the scene $\mathbf{s_i} \in \mathbb{S}$. |
| $\mathbf{z_{cls}}$ | Randomly initialised token in the same dimension as $\hat{\mathbf{t}}$. |
| $\mathcal{PE}$ | Positional embedding. |
| $\mathcal{PE}_{pos,2_i}$ | Sine function component of positional embedding. |
| $\mathcal{PE}_{pos,2_{i+1}}$ | Cosine function component positional embedding. |
| $d_{model}$ | Common dimension for both spatial and temporal tokens. |
| $\mathbf{x}$ | High-resolution frame vector sampled from the center of $\mathbb{S}$. |
| $\hat{x}$ | Spatial embedding token. |
| $\mathbf{Q}$ | Query matrix. |
| $\mathbf{K}$ | Key matrix. |
| $\mathbf{V}$ | Value matrix. |
| $Norm$ | Normalisation function. |
| $\lambda$ | Hyper-parameter to scale the influence of the temporal network. |

## Introduced in Chapter 5

| | |
|---|---|
| $\mathbb{Y}$ | Set of ground truth action instances in the input video $\mathbb{V}$. |
| $n$ | Total number of action instances in a given video. |
| $C$ | Total number of predefined action categories. |
| $s_i$ | Starting time of the action instance. |
| $e_i$ | Ending time of the action instance. |
| $a_i$ | Action category or label. |
| $c$ | Total number of predefined action categories. |
| $y_i$ | Action instance defined by its start time, end time, and action category from set $\mathbb{Y}$. |
| $\mathbb{X}$ | Visual feature set. |

| | |
|---|---|
| $\mathbb{A}$ | Audio feature set. |
| $\mathbb{F}$ | Any feature set. |
| $\mathbb{F}'$ | Downsampled feature set. |
| MaxPool | Max-pooling operation. |
| $\mathbf{x}'$ | Downsampled video feature. |
| $\mathbf{a}'$ | Downsampled audio feature. |
| $\mathbf{P_X}$ | Cross-modal projection with video as query. |
| $\mathbf{P_A}$ | Cross-modal projection with audio as query. |
| Conv1D | 1D convolution operation. |
| $\hat{\mathbf{o}}_t^l$ | Output of feature pyramid layer $l$ at instant $t$. |
| $\hat{\mathbf{c}}_t^l$ | Classification score at instant $t$ in layer $l$. |
| $\hat{\mathbf{d}}_{st}^l$ | Predicted start time deviation at instant $t$ in layer $l$. |
| $\hat{\mathbf{d}}_{et}^l$ | Predicted end time deviation at instant $t$ in layer $l$. |
| $\sigma_{IoU}$ | Temporal Intersection over Union (IoU) between predicted segment and ground truth. |

## Introduced in Chapter 6

| | |
|---|---|
| $\mathbb{P}_k$ | Set of learnable prompts for action category $k \in \mathbb{Y}$. |
| $\boldsymbol{f}_{\text{clip}}(\cdot)$ | Encoding function from a pre-trained CLIP model. |
| $\mathbf{u}$ | Distribution of video features. |
| $\mathbf{v_k}$ | Distribution of prompts for action category $k$. |
| $\boldsymbol{\delta}(\mathbf{x}')$ | Dirac delta function centered at the video feature $x'$. |
| $\boldsymbol{\delta}(\mathbf{p}_{ki})$ | Dirac delta function centered at the prompt embedding $P_{ki}$. |
| $u, v_k$ | Discrete probability distributions. |
| $\mathbf{C}$ | Cost matrix for optimal transport. |
| $\mathbf{J}$ | Transport plan matrix. |
| $e$ | Entropy term for regularisation. |
| $d_{\text{OT}}(U, V_k \vert C)$ | Optimal transport distance. |

| | |
|---|---|
| $\lambda$ | Regularisation parameter. |
| $\mathbb{X}_l'$ | Set of features at level $l$ of the temporal pyramid. |
| $\mathbf{J}_l$ | Transport plan matrix at level $l$. |
| $\mathbf{C}_l$ | Cost matrix at level $l$. |
| $\mathbf{J}_l^*$ | Optimised transport plan at level $l$. |
| $\mathbf{W}_o$ | Trainable weights for the regression task. |
| $\hat{c}_t, c_t$ | Predicted and true action categories, respectively. |
| $\hat{o}_{s_t}, \hat{o}_{e_t}$ | Predicted start and end times of actions. |
| $o_{s_t}, o_{e_t}$ | True start and end times of actions. |
| $\mathbb{1}_{\{c_t>0\}}$ | Indicator function for positive samples. |

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Human perception involves combining multiple sensory inputs to construct a dynamic understanding of our surrounding environment. Each modality—vision, hearing, touch, smell, and taste—contributes uniquely, allowing us to navigate and interact with the world effectively. This sensory processing is multimodal and inherently temporal, enabling us to perceive motion and temporal patterns essential for routine and complex activities, such as driving or participating in sports.

Our cognitive abilities extend to managing temporal invariances, ensuring consistent perception of entities across diverse sensory changes. This capability recognises objects and scenarios despite fluctuations over time, providing continuity in our perception despite a changing environment. Moreover, our brains demonstrate remarkable adaptability, enhancing specific sensory capabilities when others are diminished [185].

This complex combination of adaptive multimodal and temporal processing is crucial for understanding and learning from multi-media, especially videos. This medium inherently combines various modes of information, including visual scenes, spoken words, and background sounds. In machine learning, replicating this human-like understanding in systems is known as 'video understanding'. This task goes beyond interpreting static images to include analysing sequences of frames and integrating multiple data types to derive specific information from videos.

Achieving multimodal video understanding is challenging. Integrating various data types requires sophisticated models that can effectively parse and combine information from disparate sources.

For instance, accurately identifying the context of a scene in a video not only involves recognising visual elements and interpreting dialogues but also correlating them with relevant actions and settings. These tasks require advanced spatial and temporal data processing capabilities which are often resource-intensive. For some organisations, limited computational power, data scarcity, and lack of technical expertise limit the accessibility of deep learning approaches to improve their video processing systems. Furthermore, deep-learning techniques are increasingly being applied in broader research fields such as the digital humanities, where computational resources are not as widely available but where video understanding has the potential to capture novel insights and open new research directions.

In this thesis, we tackle these challenges by proposing innovative deep-learning techniques that enhance the efficiency and accessibility of multimodal video understanding systems. Our approaches are designed to minimise resource demands while maximising the utility and applicability of video analysis technologies. By simplifying the requirements for advanced hardware and specialised training, we aim to make cutting-edge video understanding tools more accessible to a broader range of users, including those in resource-constrained environments. This is crucial for enabling broader adoption and facilitating applications in the arts, healthcare, education, and public safety sectors, where timely and accurate video analysis can have transformative impacts.

## 1.1 Applications

Some current applications which utilise deep learning approaches for video understanding include:

**Content Moderation:** Automated systems are increasingly vital in filtering inappropriate content and ensuring user safety online. These systems detect harmful elements and analyse nuances in video and audio to identify subtle cues that might indicate malicious intent or inappropriate content. The integration of AI in this area helps to continually update and refine detection algorithms to keep pace with evolving content strategies used by users to skirt conventional detection methods. These methods include not only classification and detection [75] but also the use of multimodal video understanding to extract text [22] and understand context [76]. Effective multimodal content understanding networks should be able to discern between several modalities to identify inconsistencies, edited content, and malicious intent.

**Video recommendation and retrieval:** Enhanced by deep learning, video recommendation systems can analyse viewer behaviours and video content to personalise content delivery, thereby enhancing viewer engagement. These systems utilise complex algorithms to understand content granularly, including detecting themes, sentiment, and even the intent behind videos, enabling more refined and relevant recommendations. As styles and genres can diverge substantially over time, meta-data is becoming more unreliable for distinguishing between entertainment content. Therefore, multimodal video understanding systems are needed to discern between cinematic styles, while long-form temporal reasoning is essential for classifying narratives, characters, and relationships.

**Video Captioning and Summarization:** This application is crucial in making content accessible to a broader audience, including those with hearing or visual impairments. Advanced systems can now generate precise captions and provide summaries that capture the essence of the content, which can be invaluable for educational purposes, quick reviews, or accessibility. Automated video description and captioning technologies could be enhanced with multimodal understanding to increase the detail in the video description.

**Editing and Creative Applications:** Video understanding tools can improve the accessibility of video editing and content creation tools. AI-driven tools in video editing can automatically cut, merge, or modify video sequences, drastically reducing the time and effort required by human editors. These tools leverage video understanding to interpret scenes and suggest or execute edits that enhance the narrative or aesthetic appeal of the video. More recently, generative video diffusion networks have shown remarkable potential to generate intricate and aesthetically dynamic content via prompting. Temporal and multimodal understanding can be leveraged to improve editing to perform automated style enhancement based on genre, perform audio enhancement, and automatically summarise and edit content for film trailers or advertisements. In the generative domain, multimodal understanding can be utilised more effectively in data collection and sanitisation to ensure that the content used to train generative models includes good audio-visual alignment.

**Healthcare and Assistive Applications:** Video understanding systems can assist healthcare providers in assessing patient movement, diagnostics, and predicting falls or seizures [182].

Real-time video networks also have assistive applications such as sign-language translation, where facial expressions and hand pose estimation must be combined over temporal regions.[29]

**Robotics:** Robotic systems benefit from video understanding to mimic human actions or improve interaction with their environment. For example, robots can learn complex tasks by analysing videos of humans performing them, using this data to refine their motion algorithms and interaction strategies in functions ranging from industrial assembly to personal care. Long-term temporal modelling and multimodality are essential for learning detailed tasks from videos, which involve tracking multiple objects over time and with high temporal variance.

## 1.2 Challenges

In the domain of video understanding, two critical challenges are long-term temporal understanding and multimodal fusion.

### 1.2.1 Temporal Understanding

In video understanding, temporal modelling deals with the challenge of effectively analysing the sequence and timing of events within video data. This aspect is particularly complex due to the high frame rates and long durations typical of video files, which produce vast amounts of data.

Consider a typical scenario where video is captured at 16 frames per second. A minute of video would then accumulate 960 frames. Processing every single frame with high-dimensional data analysis would be computationally prohibitive for most systems, especially at scale. To manage this, an initial step often involves reducing the frame rate or compressing consecutive frames into a more compact representation.

A popular method for this compression is the application of 3D Convolutional Neural Networks (3D CNNs) [30], designed to capture the temporal patterns across frames while simultaneously processing the spatial information. By applying a 3D CNN to blocks of 16 frames, we can reduce these into a single, denser feature representation, significantly decreasing the amount of data that needs to be processed.

Once these features are extracted, a common technique includes temporal aggregation of the features. This process is essential for synthesising time-based information into a coherent analysis of what the video content represents over time. Methods like recurrent neural networks (RNNs) [173] and extended short-term memory networks (LSTMs)[53] have traditionally been used for this purpose because of their ability to handle sequences of data. More recently, attention mechanisms and transformers have gained popularity for their ability to weigh the importance of different segments of the video data without the constraints of sequential data processing inherent to RNNs and LSTMs [72, 154, 88, 189, 213, 9].

An example application of this technology is temporal action localisation within videos [40, 258], where the objective is to identify specific actions and their timings. For instance, detecting when a person starts running in a sports analysis video involves recognising the change from stationary to moving. Temporal modelling helps not only in pinpointing the start and end of the action but also in understanding the progression and dynamics of the motion.

The primary challenge in temporal modelling is the trade-off between accuracy and computational efficiency. More detailed temporal analysis requires more data, which can slow down processing and increase costs. A further challenge is temporal invariance, which involves identifying actions in videos over various temporal speeds, directions, and contexts, impacting the performance of automated systems in consistently recognising actions and events. Humans excel at interpreting these variations naturally, for example, understanding whether a car is starting to move or stopping regardless of how fast the video plays. Achieving this level of comprehension in video understanding networks necessitates innovative computational approaches.

One such approach is the development of SlowFast networks, which handle videos at multiple temporal resolutions to detect rapid and gradual actions effectively [62]. Similarly, Temporal Segment Networks offer another method by segmenting videos into smaller parts, thereby enhancing the model's ability to capture essential temporal dynamics within each segment [213]. However, these can only handle short video segments and have only been applied to simple video classification tasks. This work introduces several innovative methods to improve spatiotemporal understanding over long video segments and improve temporal invariance using feature pyramid networks [126] over multiple modalities.

### 1.2.2 Multimodality and Data Heterogeneity

Videos include visual content, audio signals, and textual metadata, making their analysis complex. Integrating these modalities requires innovative approaches to ensure systems can efficiently understand and leverage all available information.

The primary challenge lies in synchronising and meaningfully integrating these modalities. Each data type may require different pre-processing methods, feature extraction techniques, and learning algorithms, which can complicate the architecture of machine learning models. For example, merging audio cues with visual data can significantly enhance understanding of a scene or an event but requires sophisticated alignment techniques to match audio segments with corresponding visual frames accurately.

Moreover, heterogeneity in data also extends to variations in quality, resolution, and format, which can affect the performance of the learning algorithms. For instance, low-resolution videos offer less visual information for algorithms optimised for high-definition data, thereby necessitating adaptive methods that can effectively handle a variety of data qualities.

Addressing these challenges involves developing robust multimodal fusion techniques. These techniques can range from early fusion, where all modalities are combined at the data level, to late fusion, where each modality is processed separately, and the results are projected at the decision level. Hybrid approaches are shared, where features from different modalities are concatenated or integrated at various processing stages.

The goal is to create a unified representation that encapsulates the strengths of each modality while compensating for their weaknesses. Deep learning offers promising avenues for achieving effective multimodal fusion with techniques that can learn to extract and combine relevant features automatically, adapting to the importance of the data involved.

Throughout this work, we present several methods to improve multimodal extraction and fusion in the context of specific applications relevant to video understanding tasks and real-world applications. These include multimodal spatiotemporal fusion, gated fusion over various temporal resolutions, and collaborative gating.

## 1.3 Problem Statements and Solutions

This thesis aims to innovate several machine learning methods to address these two core challenges. We strive to exploit multimodal features via novel fusion strategies better to enhance the fine-grained nature of video content in a computationally efficient and generalisable way to be applied to long videos while improving performance over single-modality solutions. As we will see, simply adding additional data modalities can harm both performance in terms of accuracy and computational overhead. To benefit from this additional data requires application-specific approaches and fusion methodologies. In this section, we propose four application-specific problem statements under the critical challenges of efficient multimodal fusion and temporal understanding and provide a brief overview of our solutions to these challenges, which will be covered in the following chapters.

### 1.3.1 Multimodal Fine-Grained Semantic Content Retrieval and Clustering

Current video retrieval and categorisation methods depend heavily on metadata or simple visual features like RGB data. These approaches typically do not capture the intricate stylistic nuances of video content, making it challenging to accurately classify and differentiate between various sub-genres and stylistic categories within broader genre labels. For example, what is classified as 'Action' might include vastly different content in diverse cultural contexts, such as between Los Angeles and Telengana viewers. Furthermore, in many sectors, generating specific labels for data may not be possible due to cost, time, or volume. The core challenge here is to effectively break down video content into discernible stylistic elements that reflect its true nature without extensive labelling. Subsequently, these elements must be clustered to acknowledge their specific cultural and contextual differences. This task requires developing sophisticated multimodal fusion techniques that integrate and interpret diverse data types—beyond basic visual cues—to achieve a more refined and contextually aware categorisation of video content.

**Solution**

In our approach, detailed in Chapter 3, we extract several modalities from video and design a collaborative gating mechanism to identify the best method for fusing these features for a prior goal of genre classification in a supervised setting with limited labels. We fine-tune the

network in a self-supervised setting to identify sub-genres and stylistic variations within these clusters. This constructs a fine-grained latent embedding space where genres are decomposed into specific stylistic elements without additional labelling. We collect a diverse dataset of film trailers covering 100 years of cinema to achieve this goal. Through this process, we can construct a recommendation and retrieval system that uses all modalities in the temporal content to generate more stylistic and fine-grained recommendations than metadata. Significantly, we reduce the computational requirements of such a method by utilising pre-trained foundational networks to obtain feature embeddings for each modality.

### 1.3.2 Efficient Temporal Processing of Long Videos

As highlighted in the introductory chapter, handling the temporal dimension of extensive video data presents significant computational challenges. Although pre-extracting temporal features from videos can yield computational benefits during online processing, the ability to model extended temporal relationships inherently restricts the maximum length of video that can be effectively analysed. Moreover, while these pre-extracted temporal features expedite processing, they typically lack the descriptiveness of spatial RGB features, which are crucial for tasks requiring fine-grained analysis such as object recognition or action localisation.

Additionally, it has been demonstrated that self-attention network architectures, such as Transformers, often require extended training periods. This extended training time can be attributed to their lack of inductive biases, which traditional convolutional neural networks naturally possess. Inductive biases help guide the learning process, making the training more efficient. In the case of Transformers, the absence of these biases means that the model must learn to identify relevant features and their temporal relationships solely from the data or via a positional embedding, which is more computationally demanding.

To address these challenges, more sophisticated methods for temporal feature extraction that balance descriptive power and computational efficiency must be developed. Innovations could include hybrid models that integrate the strengths of convolutional architectures and Transformer-based models, leveraging the former's local processing capabilities and the latter's global context awareness. This approach could reduce the computational overhead while retaining or enhancing the ability to capture complex temporal dynamics in video sequences.

**Solution**

We introduce a simple and efficient temporal fusion mechanism in Chapter 4 that exploits inductive bias from a fine-tuned RGB CNN and fuses short snippets of pre-trained temporal CNN features. Adaptive fusion via a transformer network allows the network to use RGB features for more specific spatial features while gaining long-term reasoning ability from the temporal features. Combining convolution with the transformer for temporal modelling allows us to exploit inductive bias for more efficient training while accommodating longer video sequences. We demonstrate the network architecture's effectiveness on several downstream applications, such as speaker identification, relationship understanding, and scene understanding.

### 1.3.3 Audio-Visual Fusion

Audio information is vital in various video understanding tasks, offering unique insights not present in visual data alone. While captioning and audio-description primarily utilise audio for narrative content, action localisation demands a more fine-grained integration of audio-visual features. Audio cues are essential for accurately identifying the timing of specific actions or for identified activities occurring off-camera. However, audio content may also provide unhelpful and distracting signals in the case of edited videos where audio is misaligned, replaced, or actions in which audio markers are non-descriptive of the action taking place. This complexity necessitates advanced modelling techniques that can discern and appropriately weigh the relevance of audio signals in conjunction with visual data to enhance the accuracy of action localisation.

**Solution** To further refine our approach to the computational and analytical challenges in modelling temporal video data, our multi-resolution strategy introduced in Chapter 5 incorporates a cross-attention gating mechanism that enhances the integration of audio-visual features. This mechanism is deployed across multiple hierarchical layers within a feature pyramid network. Each pyramid layer processes different audio and visual data resolutions, allowing the network to focus on relevant features at each scale adaptively. The cross-attention gating mechanism serves as a dynamic selector, pinpointing which audio cues are most pertinent to the visual data at every temporal level. For instance, at finer resolutions where visual details are more pronounced, the mechanism can identify and amplify pertinent audio features that correspond to visible actions,

enhancing the accuracy of action localisation. This method enables audio to enhance video understanding tasks without requiring audio-specific labels, reducing the annotation burden.

### 1.3.4 Prompting Video Models

**Problem Statement:** Being able to query and retrieve videos using natural language queries opens a new avenue for future video editing, security, and health applications. Barriers to these applications include the lack of well-annotated datasets and the computational inefficiencies of training such systems. One approach is to leverage image-text models trained via contrastive image-language pre-training and adapt them for video tasks. However, this has limited accuracy on tasks such as temporal action localisation, where we must detect distinct boundaries in the temporal domain. At the same time, the RGB features may be broadly similar over the whole video segment. Refining these features via fine-tuning with limited data will inevitably lead to overfitting models that cannot generalise to unseen tasks.

**Solution**

We introduce a novel method for few-shot temporal action localisation with learnable prompts aligned using optimal transport. The idea is to fine-tune temporal prompts to introduce temporally discriminative features in a few-shot setting. We leverage optimal transport as a prompt-video alignment methodology to reduce overfitting and encourage generalisation of the features. This reduces overfitting and ensures well-generalised prompts. The methodology introduces a paradigm for few-shot prompt learning in temporal action localisation and outperforms all current methods. Through this approach, we address critical barriers in video processing and open up new possibilities for applications that require precise interaction with video content through natural language.

## 1.4 Thesis Outline

We provide solutions to these challenges in the following chapters, organised as follows.

**Chapter 2:** We first introduce relevant literature to the problems described above. This includes an overview of existing works in multimodal fusion, efficient temporal modelling, video

recognition, and retrieval, followed by specific works related to individual contributions. This includes reviewing works on temporal action localisation, supervised and self-supervised video classification, spatiotemporal fusion, and prompt learning.

**Chapter 3:** In Chapter 3, we introduce a novel method for fine-grained genre clustering with multimodal collaborative gating. We demonstrate how to extract and combine relevant multimodal features from videos and introduce a coarse-to-fine self-supervised training method for improving video retrieval and recommendation. This chapter also introduces our dataset MMX-Trailer, and we demonstrate some real-world applications using the methodology, including a semantic video-clustering tool and a video recommendation engine.

**Chapter 4:** The following chapter examines the problem of processing long videos with pre-extracted temporal features. We show how to fine-tune a lightweight RGB CNN encoder with temporal modelling via a two-stream slow-fast temporal transformer network. By fusing static frames with temporal data, we improve downstream tasks such as speaker verification, character relationship classification, and video retrieval on long videos using limited computational resources.

**Chapter 5:** Chapter 5 presents a solution for audio-visual fusion for temporal action localisation. We introduce a multi-resolution network which can improve temporal discrimination via gated cross-attention. The resulting gating network learns to amplify audio segments that are useful for the localisation objective while suppressing audio signals which are misaligned or irrelevant.

**Chapter 6:** In Chapter 6, we propose a lightweight prompt-learning method for multimodal temporal action localisation and retrieval. To ensure computational efficiency and greater generalisation capability in the network, we introduce optimal transport as a prompt-to-video alignment mechanism and evaluate the approach in a challenging few-shot scenario. We outperform existing methods for few-shot temporal action localisation while reducing the complexity and computational demand compared to existing methods.

**Chapter 7:** Chapter 7 concludes the research with a summary of the project's contributions and future directions for this research, including applications in generative media, content moderation, and video retrieval.

# Chapter 2

# Literature Review

In this chapter we provide a literature review covering some key historical works from the development of the video understanding field. Following this historical overview we provide a more detailed examination of recent works that are relevant to the challenges and solutions outlined in the introduction.

## 2.1 Introduction to Video Understanding

Over the past thirty years, video understanding has evolved dramatically, progressing from simple statistical models for recognizing actions to advanced deep learning techniques capable of analyzing and generating videos using multiple types of data. Each new approach has improved upon the limitations of earlier methods, creating a diverse and comprehensive field. This chapter reviews the historical development of video understanding techniques, focusing on the challenges faced by each generation and how newer methods addressed these challenges.

### 2.1.1 Early Statistical Methods for Video Understanding

One of the first models used for video understanding was the Hidden Markov Model (HMM). HMMs are mathematical models that use probabilities to represent systems where the actual state is not directly observable. Yamato et al. [235] used HMMs to recognize human actions by modeling sequences over time. However, HMMs had significant drawbacks, such as fixed

transition probabilities between states, making it difficult to capture the variability of real-world actions. Additionally, their reliance on the Markov property, which assumes that future states depend only on the current state, made it hard for HMMs to handle long-term dependencies.

Around the same time, Berndt and Clifford [18] introduced Dynamic Time Warping (DTW), a method for aligning sequences that vary in time. DTW was useful for recognizing gestures and analyzing motion because it could handle variations in speed and timing. However, DTW was computationally intensive, especially for long video sequences, and it required an extra classification step because it couldn't distinguish between different actions on its own.

Similarly, Optical Flow, introduced by Lucas and Kanade [136], became a standard method for extracting features for downstream processing via neural networks. Their method assumed that small regions in the image would move consistently and used mathematical optimization to find the motion parameters. Later improvements, such as the Horn-Schunck method [89] and Farneback's dense optical flow [60], increased the accuracy and reliability of motion estimation. When combined with predictive architectures, these methods helped with tasks like tracking objects and recognising actions. However, they still had challenges, such as assuming consistent motion within small areas, which led to errors with fast movements or large shifts, and being sensitive to noise and changes in lighting.

Gaussian Mixture Models (GMMs) provided another statistical approach, modeling data as a combination of several Gaussian distributions. Reynolds and Rose [171] showed how GMMs could be used for speaker identification, and Stauffer and Grimson [187] adapted GMMs for video analysis. They introduced an adaptive GMM that could update itself over time, allowing for real-time detection of moving objects against changing backgrounds, which was useful for surveillance and traffic monitoring. However, GMMs had difficulties with non-Gaussian data distributions and often needed many components to model complex backgrounds, resulting in high computational costs. The algorithm used to find the parameters, called Expectation-Maximization (EM), could be slow to converge, and GMMs were sensitive to their initial settings, often leading to suboptimal results.

To overcome these limitations, Elgammal et al. [57] proposed using kernel density estimation to better handle non-Gaussian data. Zivkovic and van der Heijden [269] further improved GMMs by

introducing a method that dynamically determined the optimal number of components, making the models more adaptable.

Kalman Filters became a robust solution for tracking objects by predicting and correcting their positions based on previous estimates. These filters were recursive, meaning they continually updated their predictions as new data came in, making them ideal for real-time tracking. Welch and Bishop [221] provided a detailed introduction to Kalman Filters. Extensions like the Extended Kalman Filter (EKF) [209] and the Unscented Kalman Filter (UKF) [104] improved their robustness for non-linear systems. Despite these advancements, challenges remained, such as the standard Kalman Filter's assumption of linearity, which limited its effectiveness for complex, non-linear systems, and its sensitivity to noise and initial estimates.

### 2.1.2 Machine Learning Advances in Video Understanding

As computational resources expanded, video classification began incorporating more sophisticated machine learning techniques. Support Vector Machines (SVMs) offered robust classification by handling non-linear data using kernel tricks [27]. However, SVMs faced challenges, such as scalability issues due to computational expense when training on large datasets and performance reliance on handcrafted features.

Feature-based methods marked a significant milestone in video classification. [118] introduced Spatiotemporal Interest Points (STIPs), which captured significant variations in motion and appearance over time. [211] advanced this by introducing Improved Dense Trajectories (iDT), which tracked densely sampled feature points to better capture motion patterns.

Robust descriptors like 3D Scale-Invariant Feature Transform (SIFT-3D) [176], 3D Histogram of Oriented Gradients (HOG3D) [116], and Motion Boundary Histograms [43] addressed the complexities of three-dimensional video data. Comprehensive libraries like Action Bank [174] broadened the scope of detectable actions in video by providing a large set of action detectors. Novel descriptors like Cuboids [52] and 3D Speeded Up Robust Features (SURF) [224] used intensity blocks and fast detection algorithms to improve feature extraction and matching in complex video scenes.

Despite their advancements, feature-based approaches required careful feature selection and extraction, and they had limited capability to model long-term temporal dependencies.

### 2.1.3 Deep Learning Revolution in Video Understanding

The introduction of Convolutional Neural Networks (CNNs) to video analysis marked a paradigm shift. [107] proposed a multi-stream CNN architecture to handle spatial and temporal features, while [183] developed a two-stream network that processed spatial and temporal data separately. However, CNNs were limited in their ability to model temporal relationships effectively.

[199] applied spatiotemporal convolutions through 3D Convolutional Networks (C3D) to capture motion directly, addressing the limitations of traditional CNNs in temporal modeling. These 3D convolutions offered significant improvements in capturing motion patterns, but the approach still had room for improvement in handling complex temporal dependencies.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) enabled sequence modeling in video understanding using CNN feature embeddings. [54] used LSTMs to capture long-term temporal relationships. Despite their effectiveness, RNNs required sequential data processing, limiting parallelization, while they performed poorly on very long sequences due to vanishing gradients.

Originally developed for natural language processing tasks, transformers have been adapted to video understanding, leveraging their self-attention mechanism to handle both spatial and temporal data without the sequential processing limitations of RNNs. This allows for parallel processing of sequences and better handling of long-range dependencies [205].

Transformer models like VideoBERT [188] and ViViT [10] utilise these capabilities in video understanding. VideoBERT, for instance, adapts the BERT architecture to video by jointly modelling visual content and corresponding textual descriptions through supervised and unsupervised learning. This dual approach facilitates understanding complex video activities by leveraging visual cues and contextual information from text [188].

ViViT further extends transformer applications by decomposing video into a space-time volume and applying factorised self-attention, which processes spatial and temporal features separately. This method allows for efficient video data handling, optimising computational resources while maintaining high accuracy in tasks such as action recognition [10].

[20] presented TimeSformer, which effectively captured global temporal relationships with self-attention. Optimized architectures like Video Swin Transformer [133] further improved

these methods by introducing hierarchical structures, reducing computational load. However, attention mechanisms are memory-intensive, especially for long video sequences, requiring optimized attention architectures to minimize computational overhead.

These transformer-based models mark a significant shift in video analysis, moving from reliance on CNNs and RNNs towards more flexible and powerful architectures capable of understanding complex, multimodal data streams.

## 2.2 Current Approaches to Research Challenges

In this section, we provide a more thorough review of the literature surrounding the two challenges of multimodal fusion for video understanding, and spatio-temporal understanding for long video.

### 2.2.1 Multimodal Fusion

Multimodal machine learning aims to build models to process and relate information from multiple sensory modalities, including audio, visual, and textual data. The field has evolved significantly, integrating various data types to enhance the richness and applicability of machine learning applications.

Initially, approaches to multimodal learning focused on simple concatenation strategies to combine features from different modalities. Notable early work included the use of Hidden Markov Models (HMMs) to align and analyse time-series data from audio and video streams for tasks like speech recognition [163]. Techniques like Canonical Correlation Analysis (CCA) [90] were also employed to find meaningful correlations between audio and visual features, improving the performance of systems in understanding speech and gestures.

As textual data was integrated, researchers began to explore the use of methods such as Latent Dirichlet Allocation (LDA) [23] for topic modeling in conjunction with audio-visual data to enhance content discovery and classification.

Deep Learning approaches to multimodal fusion can be broadly categorised into early fusion, late fusion, and hybrid approaches, each offering unique benefits and suited for different types of video analysis tasks.

**Early Fusion:** In early fusion, also known as feature-level fusion, features from audio, video, and text are combined before entering the learning model. This approach allows the model to learn from the integrated features, making it effective for tasks where the interdependence of modalities is strong. For example, early fusion has been employed in emotion recognition from videos, where cues from facial expressions, voice tone, and spoken words are integrated at the feature level to predict emotions accurately [16]. Other examples include early fusion of spatial and temporal streams for action recognition [183] and the early fusion of audio, text, and visual embeddings to retrieve relevant video clips from large databases effectively [141].

**Late Fusion:** Late fusion, or decision-level fusion, involves merging the outputs of separate models for each modality at a later stage. This technique is beneficial when each modality contains distinct information that can independently contribute to the final decision. In video sentiment analysis, separate models might analyse visual elements, spoken content, and acoustic features, with their results combined at the decision level to determine the overall sentiment [159]. Late fusion techniques have significantly enhanced video understanding by integrating outputs from separate models processing different modalities. One notable application involves combining temporal and spatial features from separate convolutional neural networks to improve large-scale video classification [107]. Another approach integrates modalities just before decision-making to enhance video action recognition [226]. Late fusion is also employed to recognise dynamic facial expressions in videos, blending outputs from multimodal setups [232], and to enhance video event detection by merging visual and textual features [244]. In video data sentiment analysis, this technique facilitates the combination of insights from video, audio, and textual analyses [159], while another application uses it for cross-modal video retrieval by combining textual and visual cues at the decision level [127]. Real-time video classification benefits from late fusion by integrating audio and visual streams [37]. The technique also aids in video-based person re-identification by merging outputs from multiple deep networks [105], and in video captioning by aggregating results from models analysing audio, visual, and textual data [87]. Furthermore, late fusion has been utilised to combine decision-level outputs from distinct modality-specific models for multimodal sentiment analysis [248].

**Hybrid Fusion:** Hybrid fusion strategies effectively leverage the strengths of both early and late fusion methods, optimising the integration of modalities for video understanding tasks. Hybrid fusion allows for flexibility in handling various modalities at different stages of the processing

pipeline, enhancing model performance and accuracy. For instance, in complex scenarios such as video summarisation, features from audio, visual, and textual data are combined early to capture comprehensive context, while decisions from these feature integrations are fused later to refine summarisation results [156]. In the domain of surveillance, hybrid fusion methods merge real-time video and audio data at an early stage, and later combine the outcomes with metadata analysis for accurate threat detection [165]. For emotion recognition from videos, hybrid approaches initially merge visual and auditory cues to capture expressive features, and textual analysis is integrated at the decision level to enhance the interpretative accuracy [16]. In automated content generation, such as video captioning, hybrid fusion is employed to preprocess visual and audio inputs together while integrating text at a later stage to generate captions that are both relevant and contextually rich [87]. Hybrid fusion is also instrumental in sports analytics, where it combines player statistics, real-time video, and commentary analysis to offer enhanced insights during live broadcasts [194].

### 2.2.2 Spatio-Temporal Modelling

As discussed, a key component of video classification, recognition, and retrieval networks involves aggregating spatial and temporal features. Early video classification works focused on non-temporal aggregation, which included clustering [99, 158, 142] or fusing [65] spatial features obtained from convolutional neural networks [115, 207, 234]. Considering that a short video will share a similar distribution of pixels over concurrent frames, these networks also perform well for video classification and object detection tasks. Naturally, fusing these output features using RNN's [173, 12, 34, 227] such as an LSTM [53, 246] improves performance by introducing temporal information. However, as discussed in [56], these model architectures align position with steps in computational time, which is inefficient at longer sequence lengths as memory constraints limit batching across sequences.

#### 3D CNN's

Later works explored extending convolution to video, inflating image CNN's via a temporal channel [61, 108, 100, 161, 201, 30] and using two stream convolutional networks to aggregate spatial and temporal information introduced via optical flow [184], or from various sampling and aggregation intervals [202, 157, 229]. 3D Convolutional Neural Networks (CNN's) are

highly effective at video classification, object detection, and action recognition tasks but are computationally intensive to train [145, 79, 231]. For example, in [200] the authors process just 16 frames at the cost of 40 GFLOPS per single pass, making the approach infeasible for sequences longer than a few seconds. To address computational overhead in training convolutional video networks, in [63] the authors present an efficient and high performance CNN for video classification which utilises a fast and slow temporal sampling stream. The slow stream can process higher resolution images and extract key spatial information over a few key frames, while the fast stream maps low resolution frames to infer temporal information. This method is highly effective at video and action classification on short 10 second clips at a low computational cost.

#### Transformers

Transformers use self-attention to model dependencies between inputs and have recently been shown to also work effectively for video classification tasks when implemented with temporal positional information [189, 135, 86, 73, 21, 216]. Unlike convolutional methods, they lack inductive bias and, as such, are not able to model finer detailed dependencies between pixel regions without extended training or the injection of positional information. While this lack of inductive bias enables transformers to achieve high levels of accuracy, it also comes at a cost to training time and data efficiency [198]. These networks are only feasible to be trained on short video snippets. Introducing inductive bias via convolution, shifting windows, and gaussian bias to transformers has shown to be effective in the image domain [245, 78, 210, 236, 123], and hybrid networks for video have also been proposed in [154, 47, 72], but only on short segments of video.

## 2.3 Application Specific Review

In this section we outline approaches to solve the applied research challenges addressed in chapters 3, 4, 5, and 6. We start by reviewing deep-learning methods applied to spatio-temporal and multimodal video clustering, recommendation, and retrieval, followed by a review of works in multimodal temporal action localisation.

### 2.3.1 Multimodal Video Clustering, Recommendation, and Retrieval

As outlined in the applications section of the introduction, effective video clustering and reccomendation methods should model both long-term temporal relationships and multimodalities. This is especially pertinent in movie retrieval and reccomendation applications, where meta-data alone is not sufficient in providing a stylistic description of a videos content. In Chapter 3 we introduce a method for the collaborative gating of multiple modalities for effective style clustering. Early approaches to this problem include extracting only low-level audiovisual descriptors before late-fusion. Huang(H.Y.) et al. [94] used two features - scene transitions and lighting. In contrast, Jain & Jadon [98] applied a simple neural network with low-level image and audio features. Huang(Y.F.) & Wang [96] used the SAHS (Self Adaptive Harmony Search) algorithm in selecting features for different movie genres learnt using a Support Vector Machine with good results. Zhou et al. [262] predicted up to four genres with a BOVW clustering technique. Musical scores have also shown to offer a useful mode for genre classification and retrieval as in the work of Austin et al. [11] who predicted genre with spectral analysis using SVMs. More recent work has utilised deep learning and convolutional neural networks for genre classification. Wehrmann & Barros [218, 219] used convolutions to learn the spatial and temporal characteristic-based relationships of the entire movie trailer, studying both audio and video features. Shambharkar et al. [178] introduced a new video feature and three new audio features, which proved useful in classifying genre, combining a CNN with audio features to provide promising results. While [179], employed 3D ConvNets to capture both the spatial and temporal information in the trailer. The ‘interestingness’ of movies has also been predicted by audiovisual features [17]. Additional features, including text and other metadata, have been combined using simple pooling in more recent work by Bonilla [32] to analyse the complementary nature of different modalities.

Self-supervised learning has also been implemented to improve the network’s stylistic discriminatory ability in the absence of style-specific labels. Self-supervised learning involves learning robust feature representations from unlabeled data by designing tasks that provide pseudo-labels. These can be obtained from the temporal [64, 220, 140] and multimodal structure of video [8, 117]. Methods exploiting the temporal consistency of video have predicted the order of a sequence of frames [64] or the arrow of time [220]. Alternatively, the correspondence

between multiple modalities has been exploited for self-supervision, particularly with audio and RGB [8, 117, 155].

### 2.3.2 Temporal Action Localization

Methods in Temporal Action Localization (TAL) can be separated into single and two-stage approaches. Where two-stage methods generate a large number of proposal segments, which are then passed to a classification head [59, 26, 83, 125, 74, 256, 125, 124, 121, 36, 59, 130], single-stage methods include the use of graph neural networks[14, 233, 249, 233] and more recently, transformers [214, 33, 192]. Recent progress in single-stage TAL has shown improvements in accuracy and efficiency over two-stage methods, combining both action proposal and classification in a single forward pass. Works inspired by object detection [170, 129], saliency detection [122], and hierarchical CNN's [240, 122, 241] all combine proposal and classification. Current SOTA methods in TAL utilise transformer-based [205] feature pyramid networks (FPN's) [250, 40, 223, 180], which combine multi-resolution transformer features with classification and regression heads.

#### Audio-Visual Fusion

Audio-visual fusion via learned representations has been explored in several video retrieval and classification tasks [58, 3, 228, 215, 151, 111, 110, 111] but audio-visual TAL has only been implemented on audio-visual events in which the audio and visual events are closely aligned [195, 13]. Concurrent works exploring audio-visual fusion in TAL have adopted two-stage late fusion approaches. Recent works have also explored audio-visual cross-attention [166] but over a single temporal resolution and without any gated fusion control.

#### Prompt Learning

Prompt learning is a methodology that introduces learnable context prompts to enhance generalization and robustness in various visual understanding tasks. Initially introduced in few-shot image recognition by CoOP [264] and later expanded upon by CoCoOp [265], prompt learning has demonstrated its efficacy in improving open-world visual understanding. This approach has also found applications in action recognition [103] and video-to-text alignment [119], facilitating tasks such as video-question answering. While the adaptation of visual-language models

for video has predominantly focused on video retrieval [49, 48, 68, 15], there have been very few works exploring how to leverage VIL models for temporal action localization efficiently. In [223], the authors use prompt learning for several video tasks, including temporal activity localization, but in a fully supervised setting. Also, in [146, 147], the authors use learnable prompts as part of a masked transformer network for classifying video region proposals.

**Few Shot Learning**

Considering that labelling or annotating videos is time-consuming, and not viable in a number of real-time contexts, few-shot learning is a relevant training paradigm for video understanding, during which one trains the network using just a limited number of labelled samples from each class. In [238], the authors introduce few-shot action-class localization in time, where a few (or at least one) positive labeled and several negative labeled videos steer the localization via an end-to-end meta-learning strategy. The strategy uses sliding windows to swipe over the untrimmed query video to generate fixed boundary proposals. In [230], the authors temporally localize an action from a few positive and negative labeled videos. They adopt a regional proposal network to produce proposals with flexible boundaries. In [149] and [223], the authors propose the challenge of zero-shot temporal action localization. However, these methods use external class scores from Untrimmed Net [212] for labeling proposals, so it is hard to evaluate their true potential in few-shot cases. In [146], the authors introduce a few-shot prompt meta-learning using additional multimodal learnable context prompts with a transformer architecture. However, they train and evaluate on a narrow $5$-way / $5$-shot meta-learning strategy and also use score fusion for the classification results.

## 2.4 Conclusion

In this chapter we have provided a comprehensive overview of the relevant literature in the video-understanding field, tracing works from the early days of statistical machine learning to recent advances in deep learning. We also introduced literature relevant to the specific chapters which follow. In the next chapter we introduce our first contribution which tackles the issue of multimodal fusion with weakly labelled data.

# Chapter 3

# Rethinking Genre Classification with Fine-Grained Semantic Experts

As discussed in the introduction, multimodal video understanding has the potential to enhance existing video recommendation systems significantly. These systems typically rely on metadata or script content, which often fail to capture the nuanced stylistic characteristics of media. Traditionally, genre has been used as part of the meta-data to efficiently describe a film's stylistic content and provide a contextual framework for viewers. However, within Film Theory, the genre is not considered a reliable descriptive label for several reasons.

Neale [153] highlights that genre labels need to be more comprehensive to encompass the diversity of content within a movie and are often only relevant to the period in which they are used. Altman [4] argues that genres are in a constant state of negotiation and change. This dynamism means that a stable set of semantic givens is continuously developed through syntactical experimentation and in response to audience or cultural shifts. Consequently, thousands of films may share identical genre labels but differ significantly in their inter-textual, multimodal, and semantic content.

Furthermore, the interpretation of genres can vary widely across geographical contexts. For instance, a Bollywood film might be categorised as a musical or romance. However, Bollywood movies possess specific multimodal stylistic elements that distinguish them from musicals produced in Hollywood. Creating additional labels for each sub-category of genres may be one

solution, but this necessitates specialist annotators. Additionally, the amount of video content may make this infeasible.

Recent machine learning-based genre classification studies have under-explored the semantic variation between genre labels [5, 218, 262, 254, 96] and do not address this issue. Furthermore, efforts have been made to avoid the issues that come with this poorly defined classification problem. In [5], the authors show how using a broader range of distribution dates within the movie dataset results in inferior classification when predicted using low-level visual features. We also find that the movie genre classification dataset LMTD-9 [218] only features movies from before 1980, which may be in response to the more fluid nature of genre representation in the last thirty years. Lastly, we find that many genre classification papers have avoided multi-label approaches [167, 96, 257], simplifying the complex relationships that exist between multiple genres [137].

Therefore, we approach genre classification as a weakly labelled problem, seeking to find similarities between the inter-textual content of movies within the genre space. To do so efficiently, we exploit expert knowledge in the form of multimodal embeddings obtained from foundation models as proposed in [132], including scene understanding, image content analysis, motion style detection and audio. Using a contextually gated approach [143] enables us to amplify modes that are more useful for multi-label genre classification and yields good results for discreet genre labelling. Then inspired by [137, 153, 4], we continue to train the model self-supervised, uniquely leveraging the similarity and differences of sub-sequences from within the trailers to identify inter-textual similarities between the movies for clustering, retrieval, and genre label improvement. This expands genre clusters by their semantic information, leading to improved clustering and retrieval without the requirement of specialist or detailed annotations.

As in other works [167, 96, 257, 218, 179], we use movie trailers as a condensed representation of the content of a movie. Since existing datasets feature a narrow range of genres and years, we also created a new 37 million frame multi-label genre dataset with pre-processed expert embeddings.

## 3.1 Methodology

This section outlines our proposed methodology for coarse classification and finer-grained clustering and retrieval. In Fig. 3.2, we present an overview of our approach. We extract audio and visual features from the input video using four pre-trained multimodal *experts*. To enable genre classification, a collaborative gating model [132, 143] learns to combine and gate these features in a supervised manner. This training has the effect of learning the most valuable combination of modes for each label. After we achieve high accuracy for multi-label classification, we encourage the network to develop fine-grained semantic clusters through self-supervised training. To achieve this, inspired by the approach of [38], we maximise the cosine similarity between sub-sequences within the trailers embedding vectors obtained from the same movie trailer (positive examples) while pushing negative sequence pairs further apart in the feature space.

Given a set of videos $\mathbb{V}$, each video is made up of a collection of sequences, $\mathbb{S}$, so $\mathbb{V} = \{\mathbb{S}^1, \mathbb{S}^2, \ldots, \mathbb{S}^n\}$. Each sequence is formed of $t$ clips, giving $\mathbb{S} = \{c^1, c^2, \ldots, c^t\}$. Ideally, the feature embeddings for all clips should have high cosine similarity as they will have the same class labels, while those from other videos with different labels should lie far apart. This work aims to create a function $\Phi$ that can map a clip $c$ from a video sequence $\mathbb{S}$, where $c \in \mathbb{S} \in \mathbb{V}$ to a joint feature space that respects the difference between clips. To construct our function $\Phi$, we rely on several pre-trained single modality *experts*, $\{\psi^1, \psi^2, \ldots, \psi^E\}$, with $E$ experts and $\psi^e$ is the $e$'th expert.

While the Motion and Scene feature extractors output a single embedding for a short sequence of frames, the Appearance and Audio features require aggregating in the temporal dimension. For Appearance features, we first concatenate the embeddings, each representing a frame in the clip and then use average pooling to obtain a single representation for the clip. For audio, we implement NetVlad [7], a method inspired by the Vector of Locally Aggregated Descriptors (VLAD) commonly used in image retrieval.

NetVLAD extends the traditional VLAD approach by introducing a learnable, differentiable mechanism for aggregating variable-length sequences of features into a fixed-length descriptor. In our audio processing pipeline, frame-level audio features are first extracted from the clip, capturing the local acoustic characteristics over time. NetVLAD then assigns these features to a

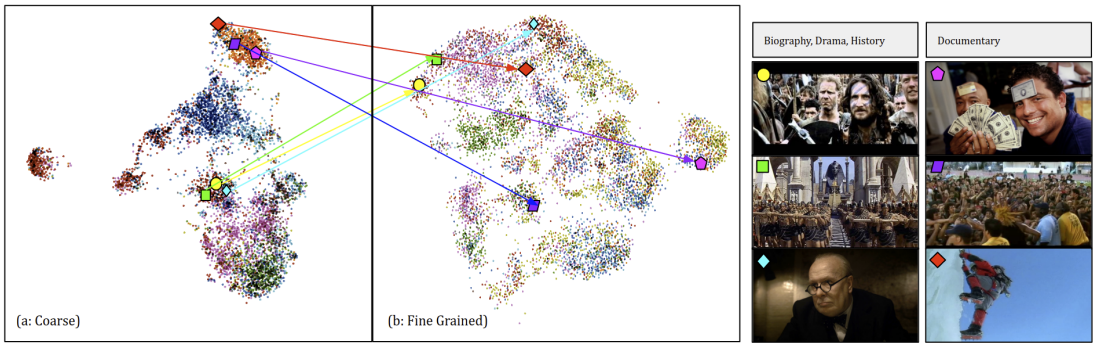



**Figure 3.1:** Self-supervised genre clustering via collaborative experts. (a) is a T-SNE plot showing the output of the coarse genre encoder network. Here, trailers that share the same three genres, 'Biography', 'Drama', and 'History', have a high cosine similarity and are well clustered, as is the 'Documentary' genre. (b) illustrates the output of our fine-grained genre model, where the model has separated the trailers, considering their multimodal content. In this example, the movie 'Darkest Hour' is pushed further away from 'Cleopatra' and 'Braveheart' as they share more similar semantic content with other large-scale historical action movies. In the second example, the three 'Documentary' trailers are pushed apart with consideration to their 'Music' and 'Adventure' inter-textual signatures.

set of learnable cluster centres, which act as prototypical sound patterns. Unlike traditional hard assignments, NetVLAD uses a soft assignment process, allowing each feature to contribute to multiple clusters based on its similarity to them.

The aggregation process involves computing the residuals between the features and their assigned cluster centres, effectively capturing the unique characteristics of the audio relative to these learned patterns. These residuals are then summed across all time frames, resulting in a fixed-length vector representing the entire audio clip. Finally, this vector is normalised to ensure consistent representation across different clips.

### 3.1.1 Collaborative Gating Unit

A two-stage process is used to learn the optimum combination of the expert embeddings for noise robustness: first, a single attention vector is defined for the $e$'th expert, and then, the expert responses are modulated with the original data.

To create the $e$'th expert's projection, we use the approach first proposed by [175, 141] for answering virtual questions. The attention vector of an expert projection will consider the

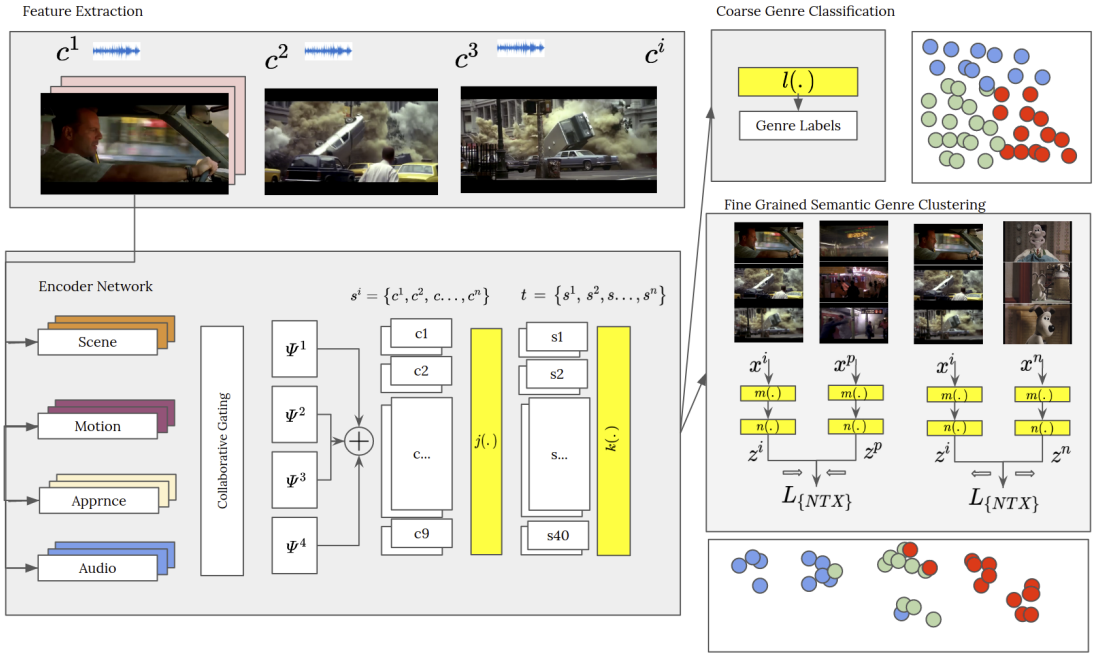



**Figure 3.2:** An overview of the approach. $c$ is a clip extracted, and $s$ is a sequence of 9 clips constructed from the concatenation of each output from the Collaborative Gating Units. The video sequences are concatenated and passed through the bottleneck MLP $\boldsymbol{k}(\cdot)$, generating the feature embedding vector. Further training for classification encourages this embedding to capture coarse genre information. After training, the whole network is fine-tuned using the self-supervised approach to encourage $\boldsymbol{k}(\cdot)$ to highlight similar inter-textual information between samples for fine-grained clustering. Dots here represent broad genres such as Action, Adventure and Sci-Fi. The fine-grained network separates the individual genres, drawing similar films together while retaining some of the broader genre information.

potential relationships between all pairs associated with this expert, as defined in Eq. 3.1.

$$\psi^e(\mathbf{c}) = \boldsymbol{h}\left(\sum_{\forall i}^{f \neq i} \boldsymbol{g}\left(\psi^i(\mathbf{c}), \psi^f(\mathbf{c})\right)\right) \tag{3.1}$$

This creates the projection between all expert embeddings $i$ and the current expert embedding $f$, where $\boldsymbol{g}(\cdot)$ is used to infer the pairwise task relationships while $\boldsymbol{h}(\cdot)$ maps the sum of all pairwise relationships into a single attention vector, and $c$ is the current clip. Both $\boldsymbol{h}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are defined as multi-layer perceptrons (MLPs) constructed of three layers. To modulate the result, we take the attention vector $\boldsymbol{\psi^e}(c)$ and perform element-wise multiplication (Hadamard product) with the initial expert embedding vector $\boldsymbol{\psi^f}(c)$ which results in a suppressed or amplified version of the original expert embedding such that:

$$\psi^f(\mathbf{c}) = \psi^e(\mathbf{c}) \circ \sigma\psi^f(\mathbf{v}) \tag{3.2}$$

Each expert embedding is then passed through a Gated Embedding Module (GEM) [144] before being concatenated into a single fixed-length vector for the clip. We capture nine clip embeddings before concatenating and passing through an MLP to obtain a sequence embedding. These sequence representations are concatenated together before passing through a bottleneck layer, which learns a compact embedding for the whole trailer.

### 3.1.2 Coarse Grained Genre Classification

The trailer embedding obtained from the collaborative gating unit can be trained with genre labels to enable classification. Given that each trailer can have up to six genre labels, a Binary Cross Entropy Logits Loss is minimised. First, the sequence embeddings $\mathbf{x}$ are summed over a trailer and then projected via an MLP $\boldsymbol{k}(\cdot)$ to produce a logits embedding. We then proceed to minimise a Binary Cross Entropy Logits Loss until convergence. With this method, it is also possible to perform genre classification on each sequence $\mathbb{S}$ by adjusting $\boldsymbol{k}(\cdot)$ so that $\mathbf{c}$ becomes the logit embedding rather than concatenating the whole sequence and performing the further projection. While the gated encoder accuracy is degraded slightly by this technique (as outlined later in the ablation studies, see Tbl. 4.4), it is possible to identify different subgenres

at a sequence level. For example, one could locate specific 'Adventure' sequences in a movie that only has the genre label 'Action'. We demonstrate this in Fig 3.8 where individual scenes are classified using the supervised approach.

In the results section later, we show how collaborative gating effectively improves the prediction task of user-defined labels in a fully supervised manner. In Fig 3.1 we can observe how similar genre labels are grouped following supervised learning, even if the style or content of the movie varies. Next we demonstrate how we can use self-supervised training to expand these coarse genre clusters into additional semantic and stylistic groups.

### 3.1.3 Fine Grained Semantic Genre Clustering

As discussed in the introduction, discrete genre labels are restrictive and only offer a broad representation of the content of a video. We aim to find finer-grained semantic content by identifying similarities in the videos' sounds, locations, objects, and motion. To achieve this, we extend the pre-trained coarse genre classification model with a self-supervised contrastive learning strategy using a normalised temperature-scaled cross-entropy loss (NT-Xent) as proposed by Chen [38], $\mathcal{L}_{NTX}$. In [38], image augmentations are used as comparative features to fine-tune the embedding layer of their classification network. The goal is to encourage greater cosine similarity between embeddings obtained from the same image while forcing the negative pairs apart. We uniquely extend this method to video by splitting each movie trailer into two equal lengths of sequences and using the embeddings of these sequences as the representation pairs $\mathbf{x}$.

$$\mathcal{L}_{NTX}(\mathbf{x}) = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{m}(\mathbf{x}_i), \boldsymbol{m}(\mathbf{x}_p)/\tau))}{\sum_{k\neq j}^{2N} \mathbb{1}_{n\neq p} \exp(\mathrm{sim}(\boldsymbol{m}(\mathbf{x}_i), \boldsymbol{m}(\mathbf{x}_n))/\tau)} \tag{3.3}$$

Here $x_i$, $x_p$, and $x_n$ are the feature representations, and $\boldsymbol{m}(\cdot)$ represents a projection head encoder formed from MLPs, $\tau > 0$ is a temperature parameter set at 0.5 and $\mathrm{sim}$ is the cosine similarity metric. $x_i$ and $x_p$ are two embedding vectors obtained from the same video described above, while $x_n$ is an embedding vector from another video. Here, the $\mathcal{L}_{NTX}$ loss will enforce $x_i$ closer in cosine similarity to $x_p$ but further from $x_n$. This process is illustrated in the overview Fig. 3.2.

Pairing each embedding vector from the video $\mathbf{s}$ with all other video embedding vectors will maximise the number of negatives. As a result, for each video, we get $2 \times (N-1)$ negative pairs — where $N$ is the number of videos in the dataset. Therefore, in training, we have mini-batch sequences, which comprise $2 \times (N-1)$ sequences $B = \{b_1, b_2, \ldots, b_{2\times(N-1)}\}$. The overall contrastive loss is computed as shown in the equation below.

$$\mathcal{L}_{CON}(B) = \sum_{i=1}^{2\times(N-1)} \mathcal{L}(i) \tag{3.4}$$

After training, the MLP projection head $\boldsymbol{m}(\cdot)$ is removed, and we use the bottleneck layer of the collaborative gating model as a pre-trained embedding projection network. The fine-tuning using a contrastive loss encourages the bottleneck layer to retain some coarse genre information while finding similar inter-textual elements in other trailers. This leads to a more diverse clustering of samples, identifying sub-label information within the original label clusters.

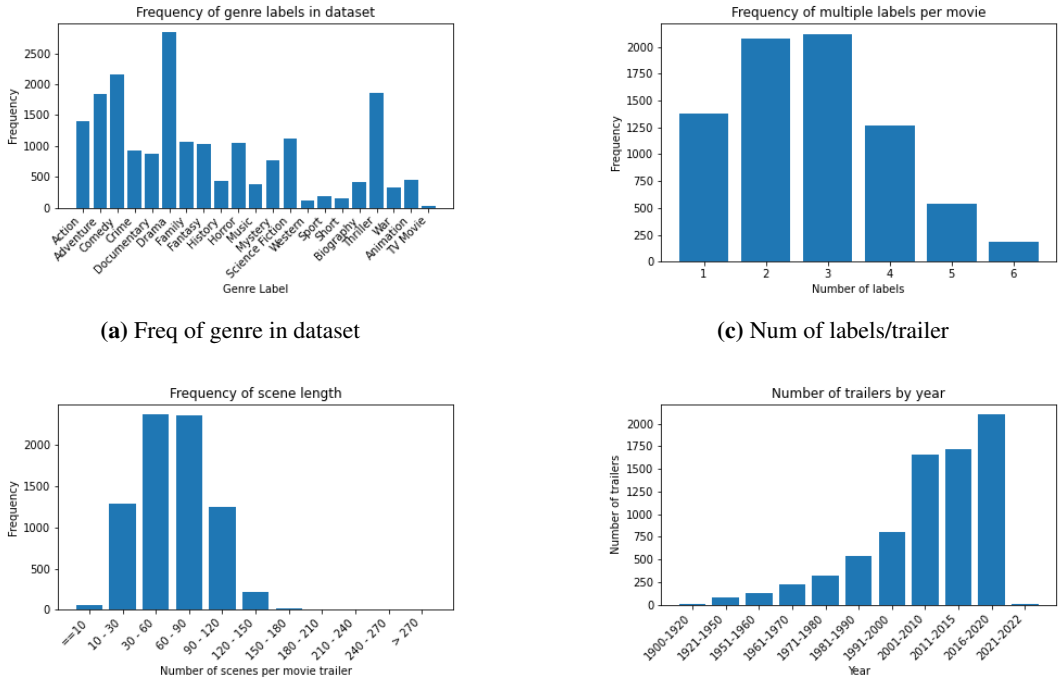
## 3.2 MMX-Trailer Dataset

There are several datasets upon which previous works test. However, capturing the scale and variability of a dataset is challenging, especially in terms of diversity of genre, size of the dataset and year of distribution. Tbl. 3.1 shows the comparison in size and labelling between recent works in genre classification.

**Table 3.1:** The details of other movie genre datasets. Our proposed dataset includes 3803 additional samples and introduces more diversity in terms of the number of genres and labels per trailer.

| Dataset | Video Source | Number Trailers | Frames | Label Source | Num. Genres | Genre/ Trailer |
|---|---|---|---|---|---|---|
| Rasheed [167] | Apple | 101 | - | - | 4 | 1 |
| Huang [96] | Apple | 223 | - | IMDb | 7 | 1 |
| Zhou [257] | IMDb+Apple | 1239 | 4.5M | IMDb | 4 | 3 |
| LMTD-9 [218] | Apple | 4000 | 12M | IMDb | 9 | 3 |
| Moviescope [32] | IMDb | 5000 | 20M | IMDb | 13 | 3 |
| MMX-Trailer-20 | Apple+YT | 8803 | 37M | IMDb | 20 | 6 |


**Figure 3.3:** Examples of the diversity of trailers within the MMX-Trailer-20 dataset. (Zoom in for detail)




**(a)** Freq of genre in dataset

**(c)** Num of labels/trailer

**(b)** Scene length

**(d)** Year of distribution

**Figure 3.4:** MMX-Trailer-20 Dataset statistics (best viewed zoomed in).

As shown, most datasets are small, with limited numbers of genre labels in terms of variability and the number assigned to a single trailer. Moviescope [32] is the closest to the proposed dataset, with three genre labels and 5000 trailers. However, we increase the number of trailers and labels per trailer while increasing the number of frames available by order of magnitude. The collection totals 8803 movie trailers drawn from Apple Trailers and YouTube, comprising 37,866,450 individual video frames. The statistics of the dataset can be seen in Fig. 3.4; for example, a wide range of genres exist, and each trailer is labelled with, on average, at least three genres, while the year of the trailers is diverse from the 1930s to the present day. We did not impose any constraints on the year of distribution during data collection, however, we observed that a substantial portion of the data comes from the 21st century, particularly from 2010 onwards. This trend can be attributed to several factors.

First, film production volume has significantly increased in the last two decades, driven by advancements in digital technology that have lowered the barriers to entry for filmmakers. Additionally, the rise of streaming platforms and video-sharing services like YouTube has revolutionised the way films are distributed and accessed, making it easier to find and collect

more recent trailers. Older films, particularly those from before the digital era, are less likely to have readily available trailers online, leading to a natural bias in our dataset toward more recent years.

Furthermore, the increased availability and popularity of online platforms from 2010 onwards have made it more convenient for both studios and independent filmmakers to distribute and promote their films, resulting in a higher volume of content being produced and disseminated. This shift has made recent trailers more accessible and easier to collect, further contributing to our collection's predominance of 21st-century data.

Every trailer is a compact encapsulation of the whole movie through a short 2 to 3-minute video, and we can collect a weak proxy of genre classification by matching the trailer to its user-generated entry on the website `imdb.com`. Users can select up to six genre labels for each trailer on IMDb. The dataset has 20 genres - Action, Adventure, Animation, Comedy, Crime, Documentary, Drama, Family, Fantasy, History, Horror, Music, Mystery, Science-Fiction, Western, Sport, Short, Biography, Thriller and War. Qualitative examples illustrating the variety of the dataset are shown in Fig.3.3.

### 3.2.1 Data Processing

The dataset is pre-processed, with scene detection performed using PyScene Detect [169], extracting individual clips from each trailer. We remove the first and last frames to mitigate poor scene detection. Audio is extracted as a mono 16bit Wav file at 16khz using FFmpeg. To compute motion frames, we use Dual $TVL1$ Optical Flow as introduced in [247] and outlined in the implementation of [191] before passing the optical flow images via the motion expert encoder. Extracted frames are also passed to the scene and appearance encoders. We partitioned the dataset into 6047 trailers for training, 754 for validation, and 754 for testing, totalling 7555 trailers. The number of trailers used for evaluation is 1248 less than the dataset as we exclude trailers with less than ten clips. This is done to maintain constant batch sizes at a minimum sequence size.

### 3.2.2 Feature Extraction

To capture the rich content of the trailer, we draw on several powerful representations present in movie trailers, *Appearance, Audio, Scene and Motion*. The ***Appearance*** feature is extracted using a SENet154 model [91], pre-trained on ImageNet [50] for the task of image classification, creating a $1 \times 2048$ embedding. The ***Scene*** feature is computed on a per frame basis from a ResNet-18 model [80] pre-trained on the Places365 [261] dataset, returning a $1 \times 1024$ embedding. The ***Motion*** of the clip is encoded via the I3D inception model [30] and a 34-layer R(2+1)D model [203] trained on the kinetics-600 dataset [30], producing a $1 \times 1024$ embedding. The ***Audio*** embeddings are obtained with a VGG style model, trained for audio classification on the YouTube-8m dataset [1] resulting in a $1 \times 128$ embedding. To aggregate the features extracted on a frame-wise basis, we average frame-level features along the temporal dimension for appearance, scene and motion embeddings to produce a single feature vector per clip per feature. For audio, we aggregate the features using a vector of locally aggregated descriptors as outlined in [8, 7]. We then average each expert feature to the same dimension of $1 \times 768$ using adaptive average pooling before passing to the collaborative gating unit described above.

## 3.3 Implementation Details

We implemented our model using the PyTorch library, and hyper-parameters were identified using coarse to fine grid search. For supervised coarse genre classification, the Binary Loss is reduced over 200 epochs using the Adam Optimiser [114] with AMSGrad [168], and with an initial learning rate of 3e-5, and a batch size of 32 samples. In the self-supervised training, we adjust the learning rate to 1e-3. We pass ten epochs of the samples through the encoder before reducing the learning rate using cosine annealing as proposed in [38]. We continue to fine-tune the network for 50 epochs. Once the semantic encoder has been fine-tuned, we remove the projection head network and then use the output of the bottleneck layer at run time.

## 3.4 Results

We evaluated our method on the MMX-Trailer-20 Dataset for genre classification and retrieval in the first stage and provided quantitative results for the second stage of self-supervised training.

### 3.4.1 Evaluation Metrics

We use the standard retrieval metrics as proposed in prior work [55, 143, 145]. Given the variance of the frequency of occurrence of the genre labels in the dataset, we employ the following metrics designed to cope with unbalanced data: $\overline{AU(PRC)}$ (micro average), $AU(\overline{PRC})$ (macro average), and $AU(\overline{PRC})_w$ (weighted average). Each measure emphasises different aspects regarding the method's performance. The $\overline{AU(PRC)}$ measure averages the areas of all labels, which causes less-frequent classes to have more influence in the results. We aim to ensure that we perform well across all categories, even for those with fewer training samples or who are more difficult to predict. $AU(\overline{PRC})$ uses all labels globally, which makes high-frequency classes have a more significant influence on the results, ensuring that we obtain overall good results across all samples in the dataset. Finally, $AU(\overline{PRC})_w$ is calculated by averaging the area under precision-recall curve per genre, weighting instances according to the class frequencies, allowing each sample to be measured independently from the whole set, and then gives us an averaged score. We also show weighted Precision ($P_w$), weighted Recall ($R_w$), and weighted F1-Score ($F1_w$); for all metrics, higher is better.

### 3.4.2 Coarse Grained Genre Classification Results

We analyse the performance of our approach both quantitatively and qualitatively for both classification and self-supervised retrieval on the *MMX-Trailer-20*, trailer dataset. Tbl. 3.2 illustrates the quantitative performance of the coarse genre prediction model, intra-genre, and the global metrics. The table also shows the random performance, which will vary according to the frequency of the genre in the dataset.

We also explore the influence of each of the individual experts on the coarse genre classification task. Individual experts are passed directly through to the first MLP, while pairs are collaboratively gated as outlined in the method. These results show that the image expert is most

**Table 3.2:** Comparison of our proposed approach with the other methods for genre classification.

| Method | no genres | no labels | $\overline{AU(PRC)}$ | $AU(\overline{PRC})$ | $AU(\overline{PRC})_w$ |
|---|---|---|---|---|---|
| Random 9 Class | 9 | 3 | 0.206 | 0.204 | 0.294 |
| Random 20 Class | 20 | 6 | 0.134 | 0.130 | 0.208 |
| VLLF [167] | 9 | 3 | 0.278 | 0.476 | 0.386 |
| AV [96] | 9 | 3 | 0.455 | 0.599 | 0.567 |
| LSTM [218] | 9 | 3 | 0.520 | 0.640 | 0.590 |
| CTT-MMC [218] | 9 | 3 | 0.646 | 0.742 | 0.724 |
| Moviescope [32] | 13 | 3 | 0.703 | 0.615 | - |
| Proposed MMX-Trailer-20 | 20 | 6 | 0.456 | 0.589 | 0.583 |

valuable for genre classification and becomes more effective when combined with motion. Using collaborative gating yields a 10% increase in basic fusion through concatenation. Audio and scene are the weakest experts for the classification task, which may be due to features that are not genre-specific, such as dialogue and external environments. All visual experts perform best in animation, most likely due to the unique style in comparison with other trailers, while audio experts perform better in comedy and sports. To identify the importance of the collaborative gating units, we compute a naive concatenation of the feature embeddings from the experts passed through an MLP layer (**Naive Concat**). This is shown to have a 10-point reduction compared to using the gating to aggregate the features, illustrating the importance of the learnt collaborative gating framework.

To attempt a comparison to other approaches, Tbl. 3.2 shows the best performance of other approaches on different datasets. Our model, *MMX-Trailer-20* uses up to 6 genre labels per sample from 20 genres, double most other approaches and will affect the random baseline, which is nearly half that of the nine genre datasets. To contextualise our method, we compare previous approaches including video low-level features(**VLLF**) [167], audiovisual features (**AV**) [96, 32], audiovisual features with convolutions over time **CTT-MMC** [218], and an **LSTM** model that uses visual feature data in a standard sequence analysis approach as implemented for comparison in [218], from the results in Tbl. 3.2 we show that our model performs better than low-level features and the LSTM model. We do not improve performance on other audiovisual approaches which fine-tune pre-trained networks in an end-to-end manner [218, 32] which use a far smaller subset of genre and labels in their older datasets.

**Table 3.3:** Coarse genre classification of the MMX-Trailer-20 dataset. Across differing expert features and combinations methods (note $(\overline{PRC}) = AU(\overline{PRC})_w$)

| Model | Actn | Advnt | Animtn | Bio | Cmdy | Crme | Doc | Drma | Famly | Fntsy | Hstry | Hrror | Mystry | Music | SciFi | Wstrn | Sprt | Shrt | Thrll | War | $F1_w$ | $(\overline{PRC})$ | $P_w$ | $R_w$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Support | 130 | 197 | 46 | 13 | 224 | 102 | 87 | 267 | 117 | 115 | 44 | 104 | 41 | 86 | 107 | 181 | 30 | 45 | 12 | 21 | - | - | - | - |
| Random | 0.29 | 0.41 | 0.11 | 0.03 | 0.46 | 0.24 | 0.21 | 0.52 | 0.27 | 0.26 | 0.11 | 0.24 | 0.1 | 0.2 | 0.25 | 0.39 | 0.08 | 0.11 | 0.03 | 0.05 | 0.318 | 0.134 | 0.19 | 1 |
| Scene [80] | 0.43 | 0.55 | 0.74 | 0 | 0.49 | 0.38 | 0.63 | 0.55 | 0.51 | 0.28 | 0.24 | 0.42 | 0.3 | 0.28 | 0.41 | 0.51 | 0.22 | 0.19 | 0.11 | 0.33 | 0.434 | 0.489 | 0.437 | 0.48 |
| Audio [1] | 0.47 | 0.51 | 0.40 | 0.10 | 0.61 | 0.38 | 0.58 | 0.55 | 0.51 | 0.37 | 0.11 | 0.34 | 0.39 | 0.30 | 0.35 | 0.55 | 0.16 | 0.15 | 0.13 | 0.12 | 0.454 | 0.449 | 0.400 | 0.537 |
| Motion [30] | 0.5 | 0.59 | 0.74 | 0 | 0.62 | 0.33 | 0.63 | 0.56 | 0.55 | 0.36 | 0.2 | 0.38 | 0.45 | 0.24 | 0.37 | 0.57 | 0.23 | 0.14 | 0.10 | 0.13 | 0.463 | 0.487 | 0.448 | 0.494 |
| Image [91] | 0.48 | 0.63 | 0.79 | 0.12 | 0.65 | 0.41 | 0.60 | 0.59 | 0.55 | 0.42 | 0.25 | 0.47 | 0.42 | 0.29 | 0.50 | 0.54 | 0.34 | 0.19 | 0.12 | 0.31 | 0.516 | 0.554 | 0.493 | 0.572 |
| Image + Audio | 0.52 | 0.63 | 0.78 | **0.15** | 0.65 | 0.42 | 0.68 | 0.6 | 0.63 | 0.46 | 0.25 | 0.50 | 0.51 | 0.34 | 0.49 | 0.59 | 0.38 | **0.28** | 0.12 | 0.42 | 0.544 | 0.558 | 0.476 | 0.65 |
| Image + Motion | 0.59 | 0.64 | 0.78 | 0 | 0.59 | 0.39 | 0.66 | 0.6 | 0.6 | 0.5 | 0.29 | 0.54 | 0.53 | 0.25 | **0.52** | 0.57 | 0.4 | 0.2 | 0.24 | 0.12 | 0.535 | 0.553 | 0.511 | 0.583 |
| Image + Scene | 0.52 | 0.61 | 0.80 | 0.12 | 0.61 | 0.37 | 0.65 | **0.62** | 0.58 | 0.49 | 0.15 | 0.51 | 0.49 | 0.37 | 0.48 | 0.56 | **0.43** | 0.26 | 0.12 | 0.46 | 0.531 | 0.539 | 0.490 | 0.600 |
| Naive Concat | 0.56 | 0.61 | 0.64 | 0.09 | 0.64 | 0.35 | 0.69 | 0.60 | 0.58 | 0.39 | 0.19 | 0.49 | 0.45 | 0.21 | 0.48 | 0.6 | 0.39 | 0.28 | 0.27 | 0.41 | 0.525 | 0.497 | 0.522 | 0.551 |
| MMX-Trailer-20 | **0.62** | **0.69** | **0.71** | 0.11 | **0.71** | **0.53** | **0.73** | **0.62** | **0.64** | **0.51** | **0.34** | **0.56** | **0.60** | **0.45** | 0.50 | **0.64** | 0.30 | 0.11 | **0.13** | **0.55** | **0.597** | **0.583** | **0.554** | **0.697** |

### 3.4.3 Fine Grained Genre Exploration

While the coarse genre classification is interesting, discreet labels are generally limited in providing a complete understanding of complex trailers. We evaluate the effectiveness of the self-supervised fine-grained genre learning by comparing the cosine similarity between embedding trailer vectors before and after being processed by the fine-grained self-supervised network. This is visualised in a T-SNE plot in Fig. 3.1, where the colours indicate the primary genre. Fig. 3.1(a) shows the learnt embedding for the coarse genre classification, where tight genre clusters are formed. Fig. 3.1(b) is after the self-supervised training of the model, where we can see how the clusters have broken up into an overlapping distribution as genres are separated depending on the multimodal content present in the trailer. We have identified three trailers (Cleopatra, Braveheart, and Darkest Hour) which share the triple genre classification of *Drama, Biography, History* (determined by the three shapes). These are correctly labelled by the coarse genre encoder, have a high cosine similarity and in Fig. 3.1(a), are spatially close in the coarse genre T-SNE plot. In Fig. 3.1(b), after self-supervised training, the trailer embeddings have higher cosine similarity to other genres. For example, Cleopatra is drawn closer to Adventure films featuring deserts and orchestral scores (Lawrence of Arabia is one example). Braveheart shares a high cosine similarity with medieval and mythological trailers featuring large-scale battles, while Darkest Hour moves towards a cluster featuring historical thrillers such as The Imitation Game. This effect is quantified in Fig. 3.5, which shows the results of the silhouette score [172] of the embedding space during the fine-grained training phase. The decreasing score shows that the coarse model's tight but restrictive genre classification is broken, and genre overlapping occurs as the training continues.

We can also show illustrative retrieval results. We provide retrieval examples in Fig. 3.6 and Fig. 3.7. In Fig. 3.6, Query 5: Trolls (2016), retrieves its sequel, Trolls World Tour (2020) and

animated family movies featuring monsters. Query 3: Mega Shark Vs Giant Octopus (2009), has the highest cosine similarity to other sea monster and environmental disaster movies such as Bermuda Tentacles (2014) and The Meg (2018). In Query 4: Seethamma Vakitlo Sirimalle Chettu (2013), we discover that the model clusters other Telugu Language Films, demonstrating the model's ability to identify a cultural context within genre clusters as opposed to the coarse classifier which simply returns films of the same genre. However, we also find Bridget Jones' Diary (2001) within this cluster, suggesting that the cluster has not been completely isolated from other romantic comedies. These results demonstrate greater depth and nuance than the coarse genre classifier retrieval.

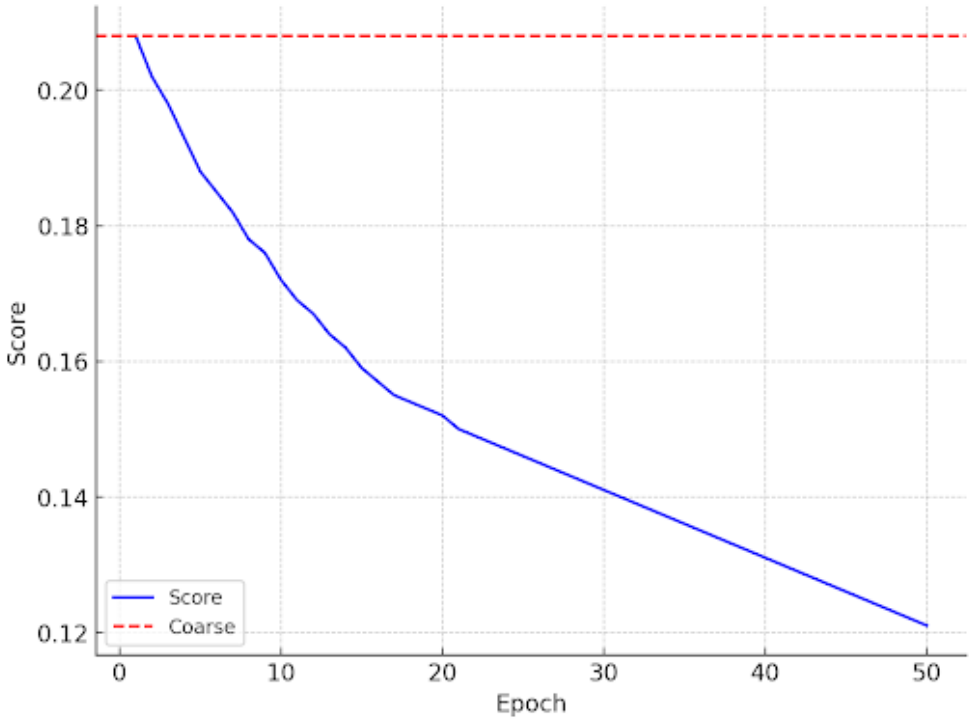



**Figure 3.5:** Silhouette score [172] of the coarse encoder output and then the following 50 epochs of fine-tuning to develop the fine-grained model. The decreasing score shows that the tight discrete genre clusters separate and overlap more as we fine-tune the network.

### 3.4.4 Augmentation of Genre Labels

It is also possible to use the self-supervised network to augment and improve the overall labelling of the original movie trailers, as shown in Tbl. 3.4. To test this, we compared genre labels produced by IMDb to find mislabelled examples in the IMDb dataset. We then asked the

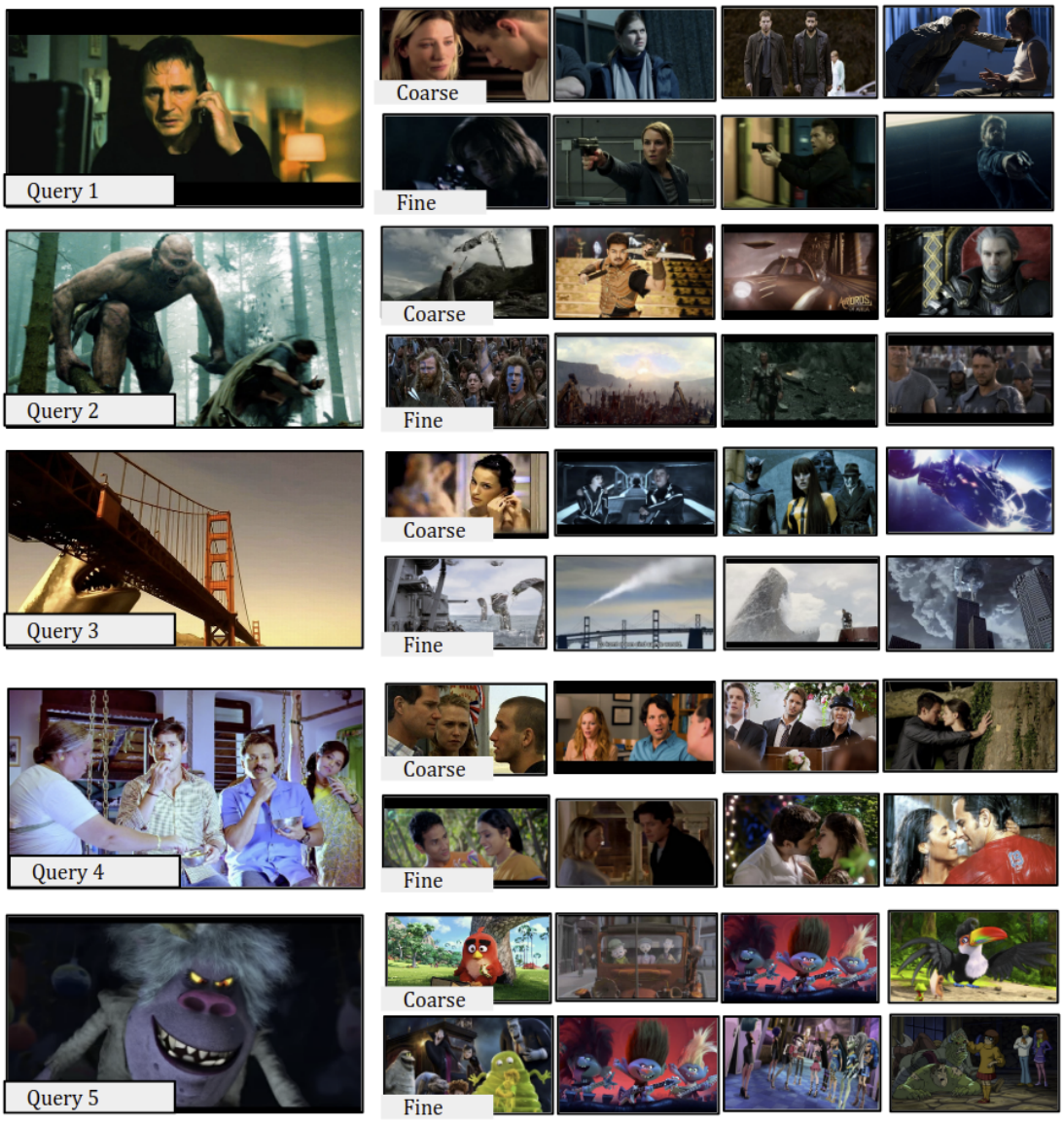



**Figure 3.6:** Retrieval results obtained from the bottleneck embedding layer before and after fine-tuning self-supervised. We can observe that the fine-grained results are much closer aligned to the themes, style, and actions in the query videos than the coarse genre label classifier.

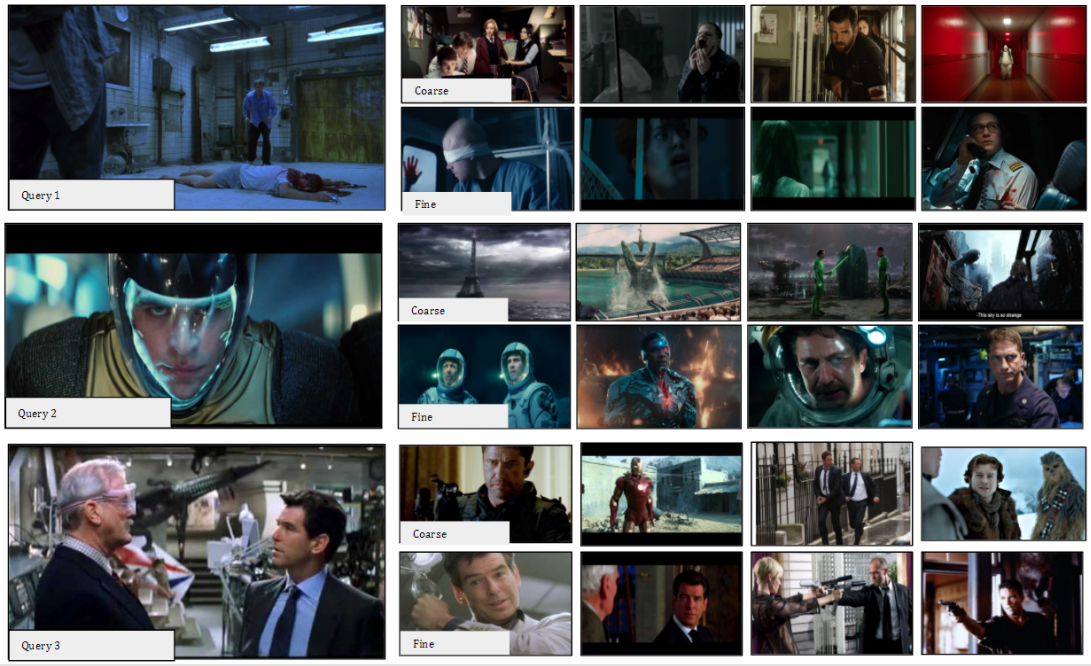



**Figure 3.7:** Additional retrieval results obtained from the bottleneck embedding layer after training for coarse genre classification and fine-tuning with the fine-grained self-supervised network. Here, the fine-grained network recommends movies from the same franchise (such as James Bond in Query 3) even though only audiovisual information is used.

network to create labels based on a sigmoid threshold of 0.30. The model extended and matched the IMDb labels and offered additional logical labels concerning the trailer. We also show how sensible labels are predicted at a scene level in Fig 3.8

### 3.4.5 Effect of Sequence Length

In [219], it is shown how capturing temporal information using 3D convolutions and LSTMs can help with genre classification. We experimented with several scene length variations to see if temporal information could be retained by concatenating scenes and sequences. First, we used the scene detection method outlined in the paper to extract individual clips before performing feature extraction and pooling to create equal 1 x 768 embeddings for every mode. After collaborative gating, each attention vector is concatenated into a sequence embedding. The concatenated sequences are then passed to the MLP before being concatenated with all other sequences to form a feature embedding for the whole trailer. In the case of self-supervised learning, we select random sequences from the same trailer for comparison.

**Table 3.4:** Examples showing additional genre labelling of movies. The original genre is the label sourced from IMDb and Predicted Genres, resulting from the proposed model. Blue indicates additional predicted labels. The additional labels are logical considering the content of the film trailer.

| **Movie** | **IMDb labelled Genres** | **Predicted Genres** |
|---|---|---|
| 101 Dalmatians II | Action, Family | Adventure, Comedy, Family, Fantasy, Animation |
| 300: Rise of an Empire | Action, Drama | Action, Adventure, Drama, Fantasy |
| Alien: Covenant | Horror, Sci-Fi, Thriller | Action, Adventure, Horror, Sci-Fi |
| Company Of Heroes | Drama, War | War, History, Drama, Action |
| Independence Day: Resurgence | Action, Sci-Fi | Action, Sci-Fi, Adventure, Thriller |
| Laws of Attraction | Comedy | Comedy, Crime, Drama |
| Leprechaun Returns | Comedy, Fantasy, Horror | Adventure, Comedy, Fantasy, Horror |
| Santa Paws 2 | Family | Family, Comedy, Adventure, Fantasy |
| The Hobbit: The Battle of the 5 Armies | Action, Adventure | Adventure, Fantasy, Action, Sci-Fi |
| The Land Before Time VIII | Family, Adventure | Adventure, Fantasy, Family, Animation |

**Table 3.5:** The effect of sequence length on classification accuracy across several metrics.

| Sequence Length | $F1_w$ | $(\overline{PRC})$ | $P_w$ | $R_w$ |
|---|---|---|---|---|
| 1 | 0.456 | 0.451 | 0.475 | 0.484 |
| 5 | 0.493 | 0.518 | 0.428 | 0.625 |
| **9** | **0.564** | **0.583** | **0.554** | **0.611** |
| 20 | 0.495 | 0.503 | 0.493 | 0.576 |

In Tab 3.5, we show the effect of sequence length on classification accuracy across several metrics. We find that longer sequences assist the model in making more accurate genre predictions, suggesting that temporal information is captured through the concatenation of scenes. However, after a sequence length of 10, we notice that accuracy decreases. We select a sequence length of 9 scenes for the model to ensure we can use as much data as possible without compromising performance.

In Fig 3.8, we can see that our model makes good predictions on individual scenes and offers reasonable guesses considering the scene's content in isolation from the whole trailer. This explains why the model performs poorly with shorter sequences. We also notice how genre predictions change throughout a trailer on a scene-by-scene basis. This is even more prevalent in modern movies, where genre fluidity is common or where other genres are referenced for narrative and stylistic effect.

### 3.4.6 Effect of Individual Experts

In Fig 3.9, we show the precision-recall curves for each label and modal expert. As might be expected, 'Animation' is the best-performing label among visual experts, with 'Comedy' performing well over both audio and visual experts. 'Documentary' is best identified by the scene and audio experts, perhaps because of the additional dialogue and use of establishing shots in 'Documentary' trailers. While we expected the audio expert to perform best on the 'Music' label, we find that image and scene experts perform just as well. This may be due to the image expert identifying instruments and the scene expert associating music with auditoriums and stadiums. The influence of different experts over the labels demonstrates the advantage of using a collaboratively gated, multimodal approach.

## 3.5 Application Examples

To demonstrate how this method could be implemented in real-world applications, we developed an embedding visualisation application and a recommendation tool that cover two real-world use cases of the methodology.

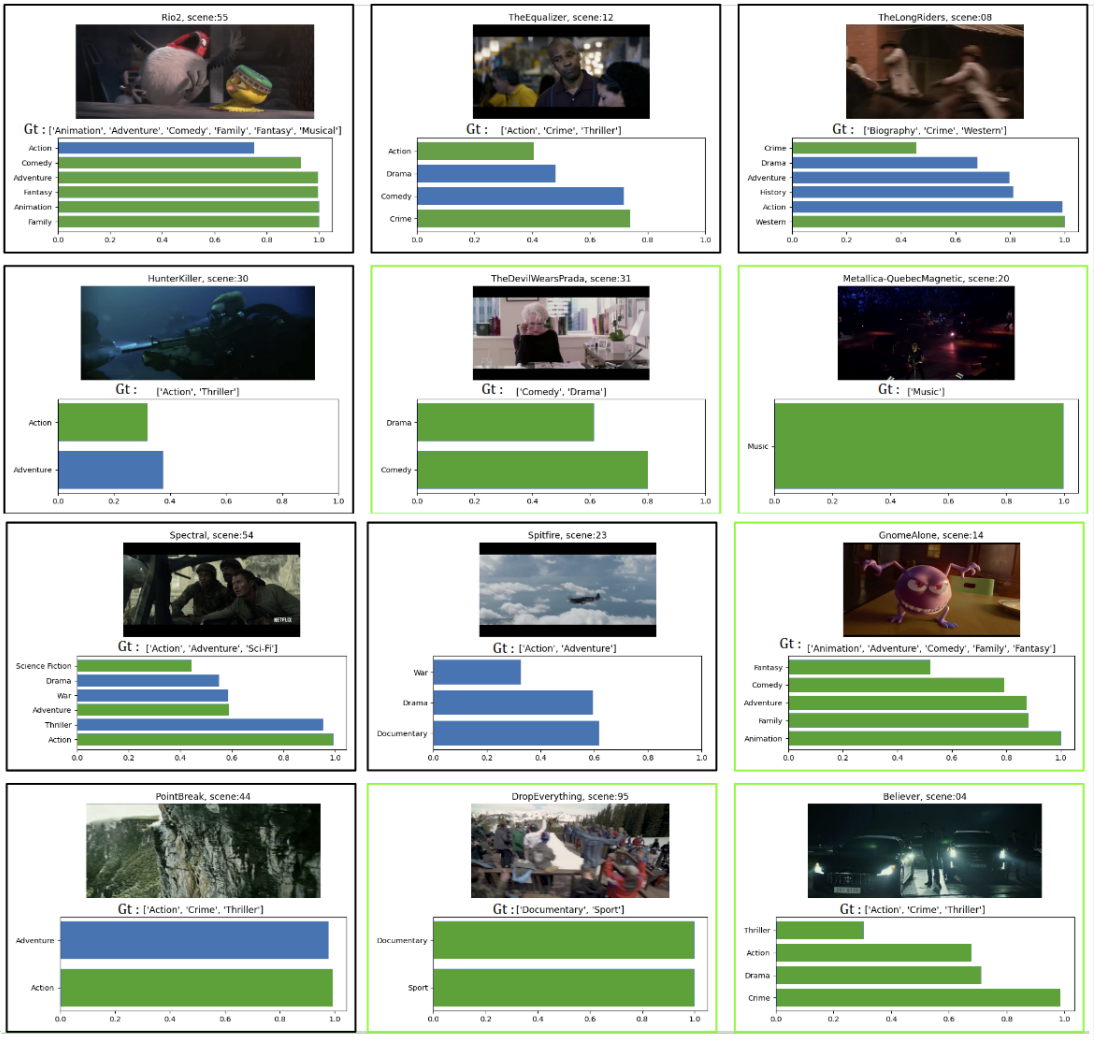



**Figure 3.8:** Representative results for multi-label classification on single scenes. Green represents results where the model predicts the correct number and labels for the scene. Black indicates where the model suggests alternative genres for the scene. We found that the model could make adequate guesses at the individual scene genre when not presented in context with the whole trailer. For example, scene 44 from Point Break (bottom left) predicts 'Adventure, Action,' which would be a good prediction given the scene's image, camera motion, and audio.

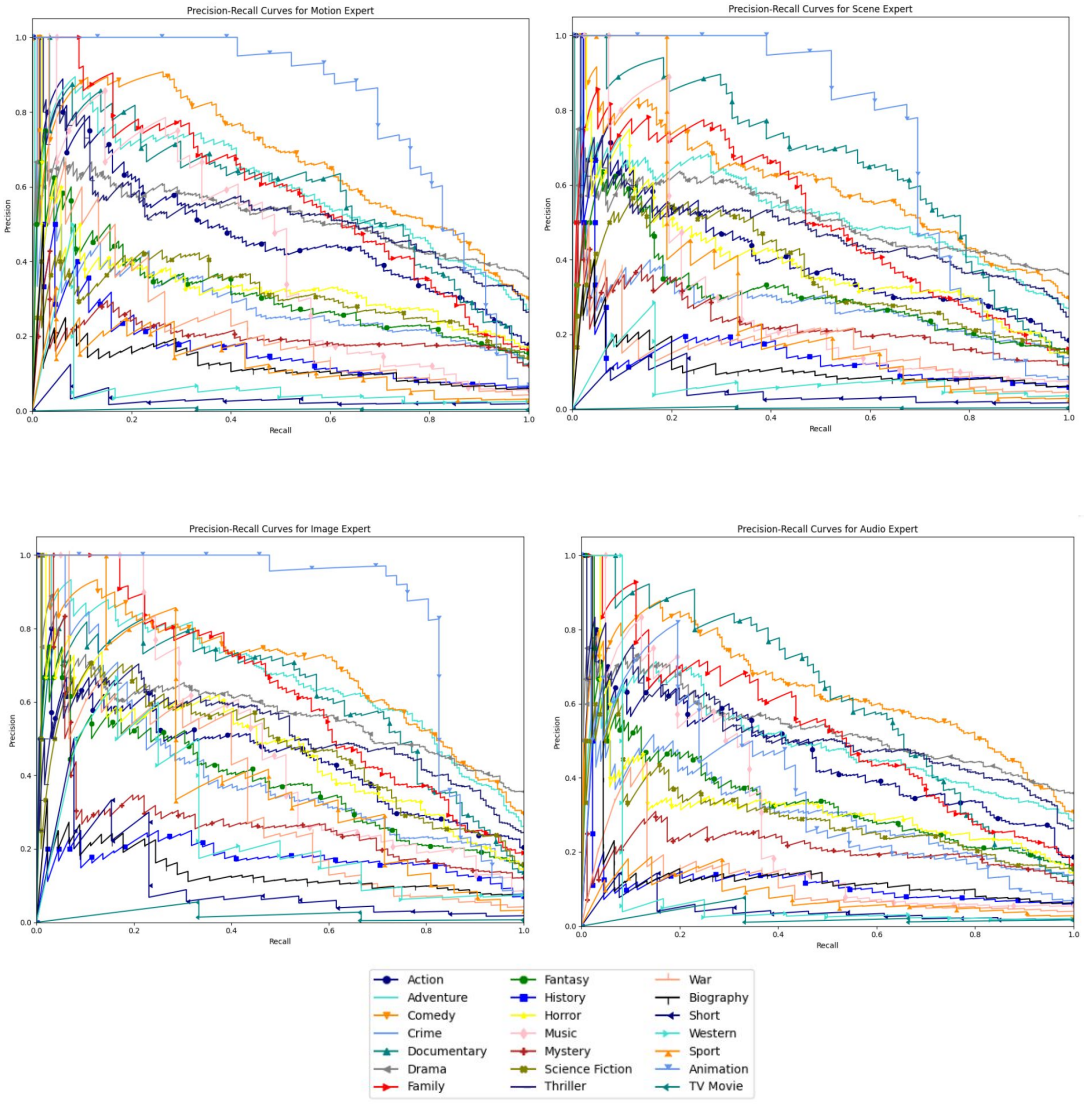



**Figure 3.9:** Precision-recall curves for each expert over all labels.

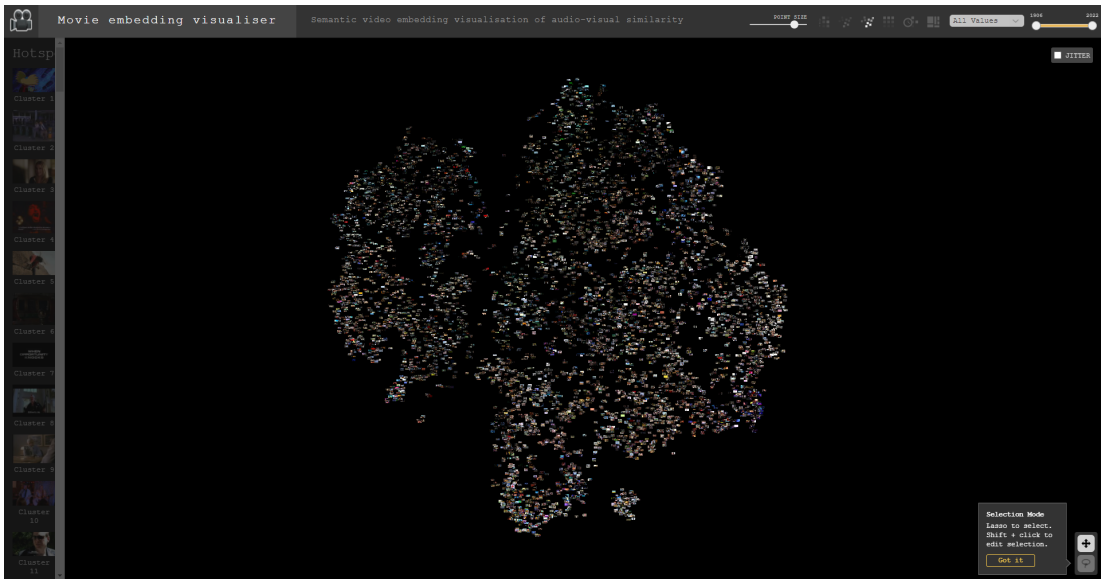



**Figure 3.10:** Semantic Video Embedding Visualisation Tool. The tool enables users to examine clustering methods such as TSNE and UMAP, select data by date periods, view videos, and find similar content using audiovisual information extracted via the method outlined in the chapter. The tool can be accessed at **https://t.ly/2l5LC**.

### 3.5.1 Style Embedding Visualisation

An overview of the tool is shown in Fig 3.10. The tool is designed for media studies, digital humanities, or film history researchers. Using our methodology, the user can interactively view both TSNE and UMAP projections of the embeddings created. Since we only use audiovisual information, this tool uniquely identifies similar stylistic themes across genres, cultures, and history. Features include:

- Ability to label and share interesting clusters found in the data.
- Users can adjust the date range to only use specific periods of film history.
- Functionality to view only films from specific genres.
- Videos can be viewed and are served dynamically via the YouTube API, or users can upload their video archives for examination.

### 3.5.2 Video Recommendation Engine

We also provide a prototype video recommendation engine that allows users to find similar movies using only multimodal audiovisual features. For the implementation, we use the Approximate Nearest Neighbours Oh Yeah (ANNOY) library [19], which efficiently indexes high-dimensional spaces to enable fast and scalable nearest neighbour searches. By leveraging ANNOY, our recommendation engine can quickly find and suggest movies with similar audiovisual characteristics. The engine processes audio and visual features extracted from the video content, combining them into a unified feature vector for each movie. These vectors are then used to populate the ANNOY index, allowing for rapid retrieval of similar movies based on user preferences. This approach ensures that recommendations are relevant and contextually rich, enhancing the user experience by presenting content that closely matches their tastes and viewing history. An overview of the interface is shown in Fig 3.11.

## 3.6 Conclusion

This chapter introduced a method for fusing multimodal video features for fine-grained video understanding applications such as style retrieval and clustering. Using collaborative gating to fuse audiovisual information enabled the network to focus on specific semantic aspects in the multimodal content of the videos while introducing the ability to differentiate between various sub-genres and categories. This method enables improved video clustering and retrieval without the need for additional annotation.

One area for improvement of this approach is the limited temporal understanding since the video segments are concatenated to represent the entire video. This also limits the video length that can be processed without heavily pooling features. In the next chapter, we introduce a method for performing several detailed video understanding tasks that rely on temporal understanding and how this can be implemented efficiently over long videos.

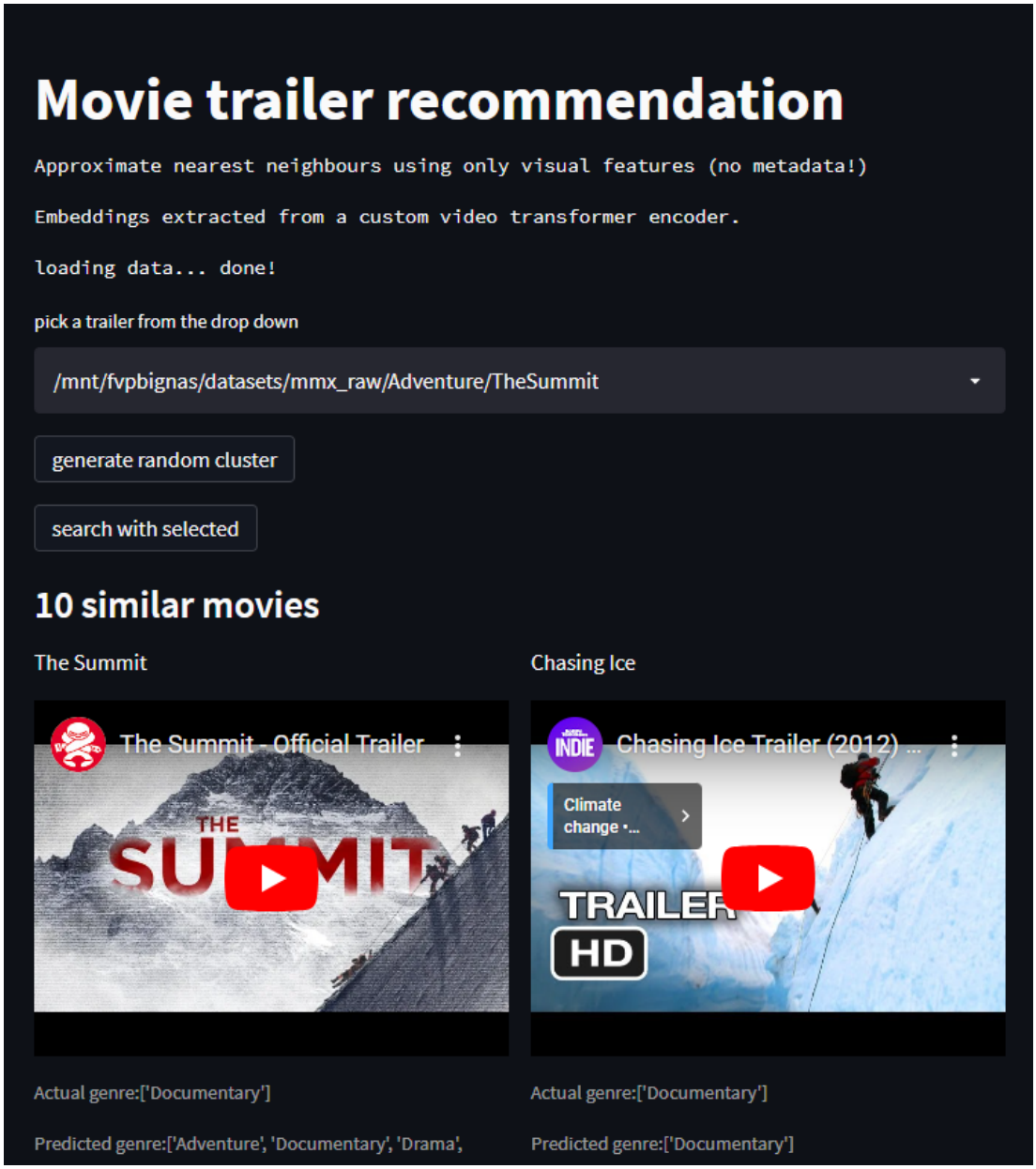



**Figure 3.11:** The interface for our prototype video recommendation engine uses the features extracted using our methodology described in this chapter. The tool can be accessed at **https://t.ly/xUDvA**.

# Chapter 4

# Two-Stream Transformer Architecture for Long Form Video Understanding

In the previous chapter, we explored methods for multimodal fusion and self-supervised video retrieval. One area for improvement in this approach is that the gated multimodal features are concatenated in the temporal dimension without any temporal reasoning for the whole video. Most video understanding tasks require in-depth temporal understanding over long media segments, especially in cases where we wish to understand aspects of a scene within the context of a whole video or we need to make some prediction on future events based on what we have seen previously. Current video recognition methods using CNN's tend to focus on short videos [139, 67, 100, 32, 95, 262, 88, 61, 161], effectively aggregating convolutional image features via late fusion and inflation [61, 108, 100, 161, 201]. Intuitively, an image can provide a good summary of a moment in time, and so it is logical that these methods perform well at classifying short videos from only a few frames [246]; however, many video understanding tasks require or can be improved with, long-term temporal reasoning [225]. The temporal fusion of features via recurrent networks [173, 53] provides one solution by aggregating frames over longer sequences of frames, but these architectures suffer from computational inefficiencies due to their recursive design. At the same time, 3D convolutional approaches also need to scale more effectively, primarily due to their high computational footprint and the linear growth of the receptive field in the temporal dimension.

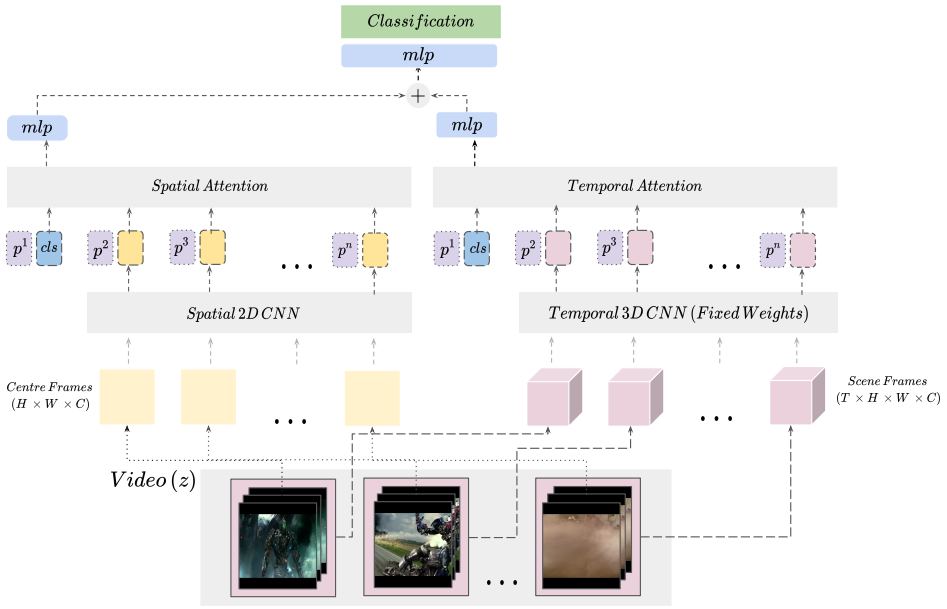



**Figure 4.1:** An overview of our approach, STAN. We encode video scenes into two feature representations using a two-stream spatio-temporal convolutional network. A transformer encoder is then used to model temporal dependencies between the tokens via an additional classification token randomly initialised. The proposed method allows us to model long-term dependencies between individual frames of long videos that feature multiple actions and environments.

More recently, transformers [51, 56] have been adapted to the video domain [9, 189, 73, 86, 21, 216], achieving state-of-the-art (SOTA) results on multiple short video tasks. However, unlike convolutional neural networks (CNNs), transformers inherently lack strong inductive biases, such as locality and translation invariance, which are crucial for efficiently processing visual data.

Inductive biases are prior assumptions built into a model's architecture that guide the learning process, enabling the model to generalise better from limited data. In CNNs, the convolutional layers are designed to exploit the local spatial structure of images by applying the same filter across different regions of the input. This ensures that patterns detected in one part of the image can be recognised in another, promoting a form of parameter sharing that significantly reduces the model's complexity. This inherent bias towards local feature extraction and translation invariance allows CNNs to learn effectively from relatively small datasets, making them highly data-efficient.

In contrast, transformers are designed with a more general-purpose architecture, relying on self-attention mechanisms that can model relationships between any pair of elements in the input,

regardless of their spatial or temporal proximity. While this flexibility enables transformers to capture long-range dependencies and complex interactions within the data, it also means that they do not assume any specific structure in the input data, such as locality in images or temporal continuity in videos. As a result, transformers must learn these patterns entirely from the data, which typically requires large amounts of labelled examples to prevent overfitting and achieve robust generalisation.

The lack of inductive bias in transformers becomes particularly problematic in the video domain, where the data is high-dimensional and temporally extended. Video data often contains a vast amount of redundant information, where frames that are close together in time are highly correlated. In the absence of inductive biases like temporal locality, transformers may struggle to learn efficiently from such data, requiring substantially more training examples to capture the relevant features across time.

Furthermore, in tasks involving long video understanding (LVU) and classification, the temporal dimension adds another layer of complexity. Transformers must process sequences that can span thousands of frames, exacerbating the data inefficiency issue. This results in models that are computationally expensive to train and deploy, with memory requirements that can easily exceed the capacities of standard GPUs, especially when dealing with high-resolution video inputs.

These challenges make transformer-based approaches less viable for domains where data and computational resources are limited. Organizations with constrained budgets, limited access to large-scale video datasets, or restricted computational power may find it infeasible to adopt these models. Consequently, while transformers offer powerful capabilities, their application to long video tasks is significantly hindered by their need for extensive data and compute, underscoring the critical role that inductive biases play in efficient model training and deployment [198].

The following question then arises: How can we leverage inductive bias from image CNNs to make video transformer networks more data—and memory-efficient for long-form video understanding tasks?

Our solution is a two-stream Spatio-Temporal Attention Network (STAN) with which we gain data and computational efficiency by introducing inductive bias via high-resolution convolution with a low-resolution temporal transformer network.

Inductive biases in neural networks significantly influence their generalisation abilities. CNNs, for instance, are imbued with a strong inductive bias towards capturing local spatial hierarchies in images through their use of local receptive fields and shared weights. This architectural design naturally suits image data, where spatially close pixels are more likely to be semantically related. On the other hand, Transformers lack such spatial biases and instead rely on self-attention mechanisms to weigh the importance of each part of the input data relative to others. This absence of inductive bias towards local structure means Transformers require substantially more data to learn patterns that CNNs capture more directly. However, Transformers offer a significant advantage in capturing long-range dependencies, a feature particularly useful for processing videos where temporal relationships across frames are crucial. By combining the spatial inductive bias of CNNs with the global receptive field of Transformers in a hybrid architecture like the proposed Spatio-Temporal Attention Network (STAN), we aim to harness the strengths of both models. This hybrid approach improves data and memory efficiency for long-form video understanding tasks by leveraging the quick, effective spatial processing of CNNs and Transformers' dynamic, detailed temporal aggregation capabilities. Integrating these biases is expected to compensate for the individual limitations of each architecture, fostering a more robust and efficient learning process for complex video understanding.

Existing methods for image classification with transformers such as [56] split images into $16 \times 16$ pixel regions encoded with a positional embedding to introduce permutation and translation invariance. This method has been extended to video by expanding these regions temporally to create 3D tokens with 2D positional embeddings [9]. Inspired by work in two-stream slow-fast convolutional video networks for short video classification [63], our approach replaces this tokenisation method with both *image* spatial and *context* temporal scene features extracted from pre-trained convolutional neural networks as shown in **??**. We then use a two-stream transformer architecture to model temporal dependencies between the scene features for classifying long videos of up to two minutes in length. By leveraging spatial and temporal features, our data- and memory-efficient method achieves competitive results on several video understanding tasks.

## 4.1 Methodology

We aim to classify long videos by splitting them into discrete scenes and extracting spatial and temporal representations for temporal fusion via a two-stream transformer encoder. We first analyse the video for average frame intensity/brightness changes using a running average over RGB video channels. We use these timestamps as scene segmentation points and uniformly sample 12 frames from each scene segment using a 3D CNN encoder to generate low-resolution temporal feature tokens for each scene. We also extract the central frame of each scene at a higher resolution and use a 2D CNN to obtain a spatial feature token. The spatial and temporal scene tokens are encoded with a positional embedding and temporally aggregated using a two-stream transformer encoder. For classification, we randomly initialise an additional token prepended to the spatial and temporal sequence of embedding tokens, which learns to model the temporal inter-dependency of the individual scenes. An overview of the sampling methodology is shown in Fig 4.2.

### 4.1.1 Temporal Encoding Token

Given a uniformly sampled set of 12 video frames from a scene defined here as $\mathbf{t} \in \mathbb{R}^{12\times H\times W\times C}$, we implement an R(2+1)d Video ResNet encoder [202] pre-trained on the Kinetics400 Dataset [30], defined as $\boldsymbol{g}(\mathbf{t})$, to encode each set of frames into a single feature embedding token $\hat{t}$, which represents the projected temporal features of the scene $s_i$ from a set of scenes $\mathbb{S}$. Using the above method, we obtain an embedding for each scene in the video to generate a set $\mathbb{Z}_{\mathbb{T}}$, representing the set of all temporal embedding vectors for the scene $s_i \in \mathbb{S}$.

$$\mathbb{Z}_{\mathbb{T}} = [\mathbf{z_{cls}}, \hat{t_0}, \hat{t_1}, ..., \hat{t_n}] + p \tag{4.1}$$

Where $\mathbf{z_{cls}}$ is randomly initiated in the same dimension as $\hat{t}_i$ and $p_i \in \mathcal{PE}$ is a positional embedding added to the sequence as described in [206] as,

$$\mathcal{PE}_{pos,2_i} = \sin(\frac{pos}{10000^{2i/d_{model}}}) \tag{4.2}$$

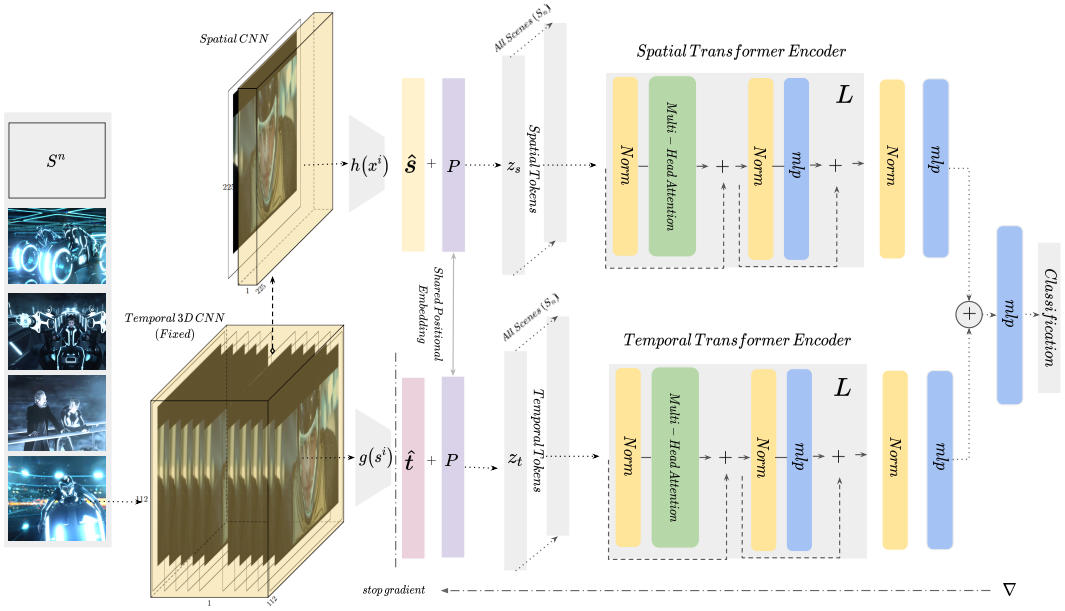



**Figure 4.2:** For each scene $\mathbb{S}$ in the video $\mathbb{V}$ we obtain a temporal ($\hat{\mathbf{t}}$) and spatial ($\hat{\mathbf{x}}$) feature embedding using 2D and 3D convolutional neural networks denoted here as $\boldsymbol{h}(\cdot)$ and $\boldsymbol{g}(\cdot)$. A shared positional embedding, $p$, is added to every scene embedding to generate the spatial tokens, $z_s$, and the temporal tokens, $z_t$. A two-stream spatio-temporal transformer with two layers ($l$) learns the dependency between the sequence of spatial tokens and the temporal tokens. Following normalisation and a linear projection, the features are fused for classification. In practice, we only fuse and classify the prepended CLS token as discussed in Section 4.1. When training the STAN-Small model, we do not back-propagate through the 3D CNN represented here by the stop-gradient line. For STAN-Large, we continue to fine-tune both convolutional encoders.

$$\mathcal{PE}_{pos,2_{i+1}} = \cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{4.3}$$

$d_{model}$ is a standard dimension for both spatial and temporal tokens. In [72, 253, 56], positional information is used to infer the relationship between cropped regions in an individual frame. As we have introduced inductive bias via convolution, we do not need to model the position of pixel regions. Instead, we extend the positional embedding method to infer the position of each short temporal sequence. This is logical for longer videos featuring a narrative composed of multiple dynamic actions and environments like those found in movies. In section 4.3, we show that this positional information benefits the model's performance on such a task.

### 4.1.2 Spatial Encoding Token

To obtain a spatial encoding token, we perform a linear projection of the high-resolution frame $\mathbf{x}$, which is sampled from the centre of $\mathbb{S}$ to a spatial embedding token $\hat{x}$ using a ResNet18[81] encoder model pre-trained on the ImageNet Dataset [50] so that $\hat{x} = \boldsymbol{h}(\mathbf{x})$. As in the case of the temporal token, we form a sequence of spatial embedding tokens $\mathbb{Z}_{\mathbb{X}}$ from spatial feature embeddings obtained via $\boldsymbol{h}(\cdot)$ so that $\mathbb{Z}_{\mathbb{X}} = \{\hat{x_0}, \hat{x_1}, ..., \hat{x_n}\}$ where $n$ is the total number of scenes in the video $\mathbb{V}$. In Eq 4.4 $\mathbf{z}_{\mathbf{x}}^{\mathbf{cls}}$ is the randomly initialised token used for classification.

$$\mathbb{Z}_{\mathbb{S}} = [\mathbf{z}_{\mathbf{x}}^{\mathbf{cls}}, \hat{x_0}, \hat{x_1}, ..., \hat{x_n}] + p \tag{4.4}$$

The positional embedding vector $p$ in Eq 4.4 is the same as defined in Eq 4.1, and as such, the positional information is shared between the spatial and temporal tokens. This ensures that positional information is consistent between the spatial and temporal streams during fusion. We now define a spatial and temporal transformer for temporal aggregation of the input tokens and then discuss techniques to fuse the output classification from both streams.

### 4.1.3 Spatial and Temporal Transformer Encoders

For the spatial and temporal transformer encoder architecture, we implement the transformer model introduced in [206] originally designed for natural language processing. As described

in [206], we generate Query ($\mathbf{Q}$), Key ($\mathbf{K}$), and Value ($\mathbf{V}$) matrices from both the spatial and temporal embedding representations where each row in the matrix represents a corresponding scene, with the first row representing our classification token. A matrix of outputs is computed as,

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\mathbf{V}\right) \tag{4.5}$$

The output matrix $\mathbf{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is summed with the input embedding matrix via a residual connection and normalised. Finally, the output is summed with a second residual connection before a classification MLP head. In practice, we only apply the final classification MLP to the temporal and spatial transformer classification tokens, which learn a representation of the input via self-attention. As such, we can discard the other rows of the output matrix and only backpropagate via the classification MLP. We use Multi-Headed-Self-Attention to model further representations in spatial and temporal domains by replicating the attention mechanism in Eq 4.5 and concatenating the heads. This process is described in detail in [206]. In [72, 253, 47] the authors use a linear layer for classification based on $\mathbf{z_{cls}} \in \mathbb{R}^d$. In our work, we linearly project $\mathbf{z_{cls}}$ to $\mathbb{R}^{d_{model}}$ for both the spatial and temporal features so we can experiment with several fusion methods, which we will describe next.

### 4.1.4 Fusion and Classification

For fusion, we normalise the embeddings and project them to a common dimension with a linear layer before classification via a three-layer MLP separated by a gated non-linearity. The loss function can be defined as a binary cross entropy loss between the targets $\mathbb{Y}$ and the scaled sum of the feature embeddings,

$$\mathcal{L}^{Fusion} = \mathcal{L}_{BCE}(\boldsymbol{h}(\text{Norm}([\boldsymbol{q}(\mathbf{z_s^{cls}}) + \lambda \boldsymbol{q}(\mathbf{z_t^{cls}})]), \mathbb{Y}) \tag{4.6}$$

Where $\lambda < 1.0$ acts as a hyper-parameter to scale the influence of the temporal network in generating the output logits and set to 0.6. $\boldsymbol{q}(\cdot)$ Is an MLP with one hidden layer and weights shared for both temporal and spatial streams. At the same time, the function $\boldsymbol{h}(\cdot)$ represents the

final MLP used for classification after fusion. To improve training time and memory efficiency, we leverage transfer learning and pre-train the 2D CNN on ImageNet [50], and the 3D CNN on Kinetics400 [30]. We present both a large and small version of the model. For the **STAN-Small**, we do not update the parameters in the temporal stream layers but back-propagate through the two-stream transformer to the 2D CNN. This reduces the number of trainable parameters by 48 Million, making the whole network trainable on a single Nvidia RTX5000 GPU with 16GB of memory. The larger model, **STAN-Large**, which back-propagates via both streams, improves performance by 11% but requires 92.5 million trainable parameters compared with 45 million for STAN-Small. In Table 4.1 we experiment with several additional methods for fusion including using collaborative gating [132] and distillation [85]. A complete experimental analysis follows in the next section.

## 4.2 Implementation Details

In this section, we discuss the implementation of the spatio-temporal Attention Network and provide further details on the network architecture, including hyper-parameter selection. We also provide further details regarding the implementations of other models used in the results section and ablation experiments.

### 4.2.1 Spatial Encoder CNN

The spatial CNN encoder is a ResNet18 Model [81] pre-trained on the ImageNet [50] dataset and implemented as described in detail in [81]. For spatial features, we randomly resize the image between $[250, 500]$ and then take a random centre crop of size $224 \times 224$. We apply additional augmentations, including flipping horizontally and vertically with a probability of $p = 0.5$ and apply Gaussian noise for additional regularisation and to prevent overfitting. During inference, we resize the shortest side of the image to $250$ pixels and then take a centre of size $224 \times 224$. For feature extraction, we replace the original classification head with a linear layer, which projects the pooled convolutional features to the size of $1 \times 896$.

### 4.2.2 Temporal Encoder CNN

The Temporal encoder is an R(2+1d) Video Classification Model [202], which decomposes spatial and temporal convolution into two steps within each convolutional block. The model is pre-trained on the Kinetics-400 Dataset [30] and implemented as described in detail in [202].

For training, we take spatially consistent random crops of size $112 \times 112$ pixels for each scene so that each frame in the scene is cropped in the same location and dimension. We also introduce random erasing of pixel regions, which covers parts of the cropped region with black squares. This improves convergence and reduces over-fitting even when we only back-propagate via the spatial encoder CNN. We sample 12 frames from each scene, which are projected via the temporal encoder to one temporal token representation of size $1 \times 896$ via a linear layer.

### 4.2.3 Spatio Temporal Attention Encoders

The two-stream encoder is constructed from a temporal and spatial transformer encoder. The transformer encoders are implemented according to the design in [206] and initialised with four heads and two layers with drop-out at $p = 0.5$. Both transformer encoders output a CLS token of shape $1 \times 896$, which is then linearly projected by a three-layer MLP with Gaussian Error Linear Units (GELU) to the size of $1 \times 128$. These output spatial and temporal features are then normalised using L2 Normalisation and summed before a final classification MLP, which linearly projects the embedding to the target shape.

### 4.2.4 Training Details

As in the previous chapter, scenes are extracted from videos in the datasets using change detection over the RGB values. This results in scenes representing a sequence of frames from a single camera shot or view. Although these scenes will be of variable length, we can presume that the action occurring within the scene is relatively static if the RGB values do not change enough for a boundary to be detected. This means we can simply extract all frames from the scene and then subsample for a reliable representation of the scene content. From each scene $\mathbb{S}$, we extract 12 equally distributed frames. The sixth frame is processed via the spatial encoder CNN, while all the frames are processed via the temporal encoder CNN.

For efficiency, we extract frames from videos offline and, inspired by [270], we use the NVIDIA DALI pre-processing and dataloading framework, which allows us to decode and transform frames on the GPU. For STAN Large, we use 4 NVIDIA-RTX-5000 GPUs and train the model for approximately 16 hours, while the STAN Small model is trained on just 1 NVIDIA-RTX-5000 GPU for 48 hours.

We use a batch size of 64 for both models, obtained by accumulating gradients across epochs. We use the Adam Optimizer [113] with $\alpha = 1e - 5$ and a weight decay set at $0.09$ for optimisation.

## 4.3 Results

We evaluate the proposed method on several long video classification tasks. As discussed in [225], video datasets have typically focused on short video tasks; therefore, following [225], we define long videos as videos which feature more than three shots or scenes, are longer than a minute in length, and in which classification relies on, or is improved by, contextual understanding of the relationship between the content of shots and their order. At first it may seem that only a minute in length is not a particularly long video however this duration is considered long in the context of video understanding tasks due to the complexity involved in processing and analyzing the relationships between multiple scenes or shots within such a time frame. Unlike short videos, which typically focus on recognizing single actions or isolated events, long videos require the model to understand and maintain context over a more extended period. This involves capturing dependencies across shots, managing the temporal dynamics of the video, and integrating this information to make accurate classifications. Consequently, even a minute-long video can present a significant challenge in terms of computational resources and algorithmic design in machine learning tasks.

To evaluate our method, we use both the **MMX-Trailer-20 Dataset (MMX)** [65], as introduced in the previous chapter, and the **Long Video Understanding Benchmark (LVU)** [225].

**LVU Benchmark**

The Long Video Understanding Benchmark [225] contains 30K videos with an average length of 120 seconds. The benchmark comprises of interaction classification, user engagement prediction, and movie metadata prediction tasks, demonstrating various requirements for long

temporal modelling. We test our approach on the content understanding tasks, including character relationship identification, speaking style, and scene recognition.

### 4.3.1 Comparative Methods

To demonstrate the effectiveness of the two-stream network, we implement several existing methods for video classification using the MMX-Trailer-20 dataset. They include extracting convolutional image features from a ResNet18 [81] for each scene and simply using average pooling for classification. We also implement this same method with features obtained from a two-stream Inflated 3D Convolutional Neural Network [100], and a SENet [92]. We also show results for comparative temporal networks, including a Temporal Pyramid Network, which extracts multiple level features from a CNN to model dynamic temporal movements in videos, and a vanilla LSTM [53] to model the temporal relationship between the feature representations from an 18 layer ResNet. More complex temporal aggregation strategies are also explored, such as the Fine-Grained Semantic [65] architecture, which uses concatenation to aggregate multimodal features as well as the effectiveness of earlier audio-visual works such as audio-visual classification using support vector machines [96] and audio VGG network as described in [2]. We also explore distillation as described in [198], using the temporal transformer encoder as a teacher network to the spatial transformer. For the LVU Benchmark task, we compare our method with those presented in the paper, including videoBERT [189], R101-SlowFast [63], and Object Transformer [225].

### 4.3.2 Metrics

As outlined in the previous chapter, the MMX dataset is imbalanced. As such, we follow other works [55, 143, 145] and use Mean Average Precision $mAP$ to evaluate the effectiveness of our classifier. To calculate the $mAP$, we average the area under the precision-recall curve per genre, weighting instances according to the class frequencies. We also show weighted Precision ($P_w$), weighted Recall ($R_w$), and weighted F1-Score ($F1_w$) in Table 4.3. With all metrics, a higher value demonstrates improved accuracy. For LVU, we compute the standard error averaged over five runs as proposed in the original paper [225] where higher values represent improved performance.

| Method | CNN Feature Extractor | Frames per Scene | Aggregation Method | $\overline{mAP}$ |
|---|---|---|---|---|
| ResNet [81] | ResNet18 | 1 | Avg Pool | 0.434 |
| SqueezeExcite [92] | SE-ResNet | 1 | Avg Pool | 0.544 |
| I3D [30] | I3D | 12 | Avg Pool | 0.487 |
| Collaborative Gating [132] | ResNet18 | 12 | Gated Unit | 0.4723 |
| S(TPN) [259] | ResNet18 | 12 | Temporal Pyramid | 0.492 |
| Fine-Grained Semantic [65] | multimodal | 16 | Concatenation | 0.583 |
| LSTM [53] | ResNet18 | 1 | LSTM | 0.596 |
| Distillation [85, 198] | ResNet18 + R2+1D | 12 | Transformer | 0.601 |
| **STAN-Small** | ResNet18 + R2+1D | 12 | Transformer | **0.640** |
| **STAN-Large** | ResNet18 + R2+1D | 12 | Transformer | **0.750** |

**Table 4.1:** Comparison of our proposed approach with existing methods for video classification using CNN feature extractors and evaluated on the MMX-Trailer-20 Dataset. We implement several feature extraction and aggregation methods to evaluate their effectiveness for the long video classification task.

| Method | Relation | Speaking | Scene |
|---|---|---|---|
| R101-SlowFast+NL [63] | 52.4 | 35.8 | 54.7 |
| VideoBERT [189] | 52.8 | 37.9 | 54.9 |
| Object Transformer [225] | 53.1 | 39.4 | 56.9 |
| **STAN-Large (ours)** | **56.25** | **41.41** | **58.33** |

**Table 4.2:** Accuracy of our approach on long video understanding tasks using the Long Video Understanding Dataset. The reader can find references and further details in [225]. We outperform current approaches for classifying conversation (speaking), character relationships (relation), and locations of scenes (scene).

### 4.3.3 Evaluation

First, in Table 4.1, we evaluate our approach against existing methods for video classification using the MMX-Trailer-20 Dataset. We demonstrate that our method outperforms the pooling of convolutional spatial features by 10%. Secondly, we show how our method improves performance on scene-level features extracted via an inflated 3D CNN [61] by 11%. Third, we compare our approach with other techniques for combining convolutional features temporally, including an LSTM [53], S(TPN) [259], concatenation [65], and collaborative gating [132]. Finally, we show results for an alternative method for two-stream aggregation using a distillation network as described in [198] and described in more detail in the following section. We outperform all existing techniques for the genre classification task.

| Model | Actn | Advnt | Animtn | Bio | Cmdy | Crme | Doc | Drma | Famly | Fntsy | Hstry | Hrror | Mystry | Music | SciFi | Sprt | Shrt | Thrll | War | $F1_w$ | $mAP$ | $P_w$ | $R_w$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Support | 130 | 197 | 46 | 13 | 224 | 102 | 87 | 267 | 117 | 115 | 44 | 104 | 41 | 86 | 107 | 30 | 45 | 12 | 21 | - | - | - | - |
| Random | 0.29 | 0.41 | 0.11 | 0.03 | 0.46 | 0.24 | 0.21 | 0.52 | 0.27 | 0.26 | 0.11 | 0.24 | 0.1 | 0.2 | 0.25 | 0.08 | 0.11 | 0.03 | 0.05 | 0.318 | 0.134 | 0.19 | 1 |
| ResNet [81] | 0.43 | 0.55 | 0.74 | 0 | 0.49 | 0.38 | 0.63 | 0.55 | 0.51 | 0.28 | 0.24 | 0.42 | 0.3 | 0.28 | 0.41 | 0.22 | 0.19 | 0.11 | 0.33 | 0.434 | 0.489 | 0.437 | 0.48 |
| VGG-Audio [2] | 0.47 | 0.51 | 0.40 | 0.10 | 0.61 | 0.38 | 0.58 | 0.55 | 0.51 | 0.37 | 0.11 | 0.34 | 0.39 | 0.30 | 0.35 | 0.16 | 0.15 | 0.13 | 0.12 | 0.454 | 0.449 | 0.400 | 0.537 |
| I3D [30] | 0.5 | 0.59 | 0.74 | 0 | 0.62 | 0.33 | 0.63 | 0.56 | 0.55 | 0.36 | 0.2 | 0.38 | 0.45 | 0.24 | 0.37 | 0.23 | 0.14 | 0.10 | 0.13 | 0.463 | 0.487 | 0.448 | 0.494 |
| SqueezeExcite [92] | 0.48 | 0.63 | 0.79 | 0.12 | 0.65 | 0.41 | 0.60 | 0.59 | 0.55 | 0.42 | 0.25 | 0.47 | 0.42 | 0.29 | 0.50 | 0.34 | 0.19 | 0.12 | 0.31 | 0.516 | 0.554 | 0.493 | 0.572 |
| Naive Concat [81] | 0.56 | 0.61 | 0.64 | 0.09 | 0.64 | 0.35 | 0.69 | 0.60 | 0.58 | 0.39 | 0.19 | 0.49 | 0.45 | 0.21 | 0.48 | 0.39 | **0.28** | 0.27 | 0.41 | 0.525 | 0.497 | 0.522 | 0.551 |
| Fine-Grained Semantic [65] | 0.62 | **0.69** | 0.71 | 0.11 | **0.71** | 0.53 | **0.73** | 0.62 | 0.51 | 0.34 | 0.56 | **0.60** | **0.45** | 0.50 | **0.64** | **0.30** | 0.11 | 0.13 | **0.55** | 0.597 | 0.583 | 0.554 | 0.697 |
| **STAN-Small** | **0.71** | 0.68 | **0.92** | **0.21** | 0.61 | **0.65** | 0.62 | **0.69** | **0.86** | 0.49 | **0.46** | **0.58** | 0.43 | 0.39 | **0.53** | 0.13 | 0.20 | **0.85** | 0.50 | **0.65** | **0.64** | **0.62** | **0.73** |

**Table 4.3:** [Genre classification performance for each genre on the MMX-Trailer-20 dataset.] Genre classification performance for each genre on the MMX-Trailer-20 dataset. We observe high-performance gains on genres where temporal information can be considered an important classifier, such as Action $+9$ and Animation $+11$. We also observe that other network architectures perform very poorly in the classification of the genre thriller while we improve accuracy by $+58$. Long-term temporal modelling performs well on this task as the content is difficult to classify when features are presented in isolation.

| Method | Samples | Parameters (Millions) | mAP |
|---|---|---|---|
| Spatial with backprop | 2000 | 28 | 0.5903 |
| Temporal with backprop | 2000 | 48 | 0.6221 |
| Spatial no backprop | 2000 | 16.5 | 0.4024 |
| Temporal no backprop | 2000 | 16.5 | 0.59 |
| Distillation Network | 2000 | 44.5 | 0.6005 |
| Gated Fusion | 2000 | 45 | 0.4728 |
| STAN-Small | 2000 | 45 | 0.6151 |
| STAN-Small | 6047 | 45 | 0.6401 |
| STAN-Large | 6047 | 92.5 | 0.7506 |

**Table 4.4:** Ablation experiments assess the network quality under a constrained data training protocol. Each model is trained using only 2000 samples, while the test length remained consistent at 754 samples. The network continues to outperform existing methods with fewer data. We also show results for individual spatial and temporal streams with back-propagation.

In Table 2, we show that we achieve SOTA results for three long-form video understanding tasks on the LVU Benchmark, outperforming other methods which also utilise transformer architectures such as VideoBERT [189] and [225], which utilises pre-trained CNN backbones plus self-supervised masked pre-training. Our architecture performs particularly well on the relationship identification task $(+\mathbf{3.1})$ despite having no prior knowledge of the domain in pre-training.

### 4.3.4 Ablation Experiments

In Table 4.4, we perform several ablation studies to assess the impact of the spatial and temporal features and the data efficiency of the model. We sampled 2000 trailers from the MMX-Trailer-20 training partition to measure the data efficiency of the entire test partition of 754. Table 4.4 shows that the model still achieves SOTA performance on the MMX-Trailer-Dataset despite being trained on only a quarter of the samples demonstrating high data efficiency. We infer that data efficiency is improved by using convolutional encoding as the transformer network only needs to map self-attention between the scene feature tokens rather than pixel localities. Furthermore, the proposed network architecture can infer translation invariance within the convolutional encoding.

In Table 4.4, we also provide further results for models that only use spatial or temporal convolutional encoders to assess the impact of propagating gradients through the convolutional encoders. We observe that back-propagating through the spatial convolutional encoder provides the most significant performance gain with the most negligible effect on the number of trainable parameters. Training both the CNN encoders end-to-end (STAN-Large) is the most effective method for achieving high accuracy. Still, it comes at a cost, increasing the number of trainable parameters by 48 million however this is still substantially smaller than existing methods such as videoBERT which has 128 million parameters. We find that using just the temporal transformer encoder and introducing a stop gradient before the convolutional feature extractor performs well but is improved with spatial features for 28 million additional trainable parameters. We also show genre-specific results in Table 4.3.

### 4.3.5 Fusion Methods

We experiment with an alternative fusion method for the classification (CLS) embedding tokens for an additional benchmark, as shown in Tab 4.1 as distillation.

In [198], the authors append a distillation token to the token sequence $\mathbb{Z}$ obtained via a pre-trained image network, which is used as a teacher output via hard or soft distillation for image classification. We extend this idea to video, using the Temporal Encoder Token as an additional distillation token, which is appended to the input sequence of the spatial encoding network. Formally:

$$\mathbb{Z}_s = [\mathbf{z}^{\mathbf{s}}_{\text{cls}}, \hat{\mathbf{x}}_{\mathbf{0}}, \hat{\mathbf{x}}_{\mathbf{1}}, \ldots, \hat{\mathbf{x}}_{\mathbf{n}}, \mathbf{z}^{\mathbf{t}}_{\text{cls}}] + \mathcal{PE} \tag{4.7}$$

where $\mathbf{z}^{\mathbf{s}}_{\text{cls}}$ is a classification token for the spatial encoding sequence, $\hat{\mathbf{x}}_{\mathbf{n}}$ are expert convolution tokens obtained via the pre-trained spatial CNN, and $\mathbf{z}^{\mathbf{t}}_{\text{cls}}$ is the temporal encoding token. The sequence $\mathbb{Z}_s$ is then passed to a single transformer encoder. We take the transformer projections of $\mathbf{z}^{\mathbf{s}}_{\text{cls}}$ and $\mathbf{z}^{\mathbf{t}}_{\text{cls}}$, where the MLP $\boldsymbol{g}(\mathbf{z}^{\mathbf{t}}_{\text{cls}})$ is used to obtain either soft targets (logits) or one-hot-encoded hard labels while $\boldsymbol{g}(\mathbf{z}^{\mathbf{s}}_{\text{cls}})$ performs classification.

While the distillation network did not perform as competitively as the primary fusion method, it did reduce training time and data efficiency as described in [198].

### 4.3.6 Qualitative Results

In Fig 4.3, we show class activation maps of the input 2D CNN encoder, shown in Fig 4.2 as $\boldsymbol{h}(\cdot)$ and the predicted genres for a given input video. We observe that the model predicts the labels correctly and learns cohesive features for the input tokens. For example, in Fig 4.3, we see that the Family genre is predicted when we have a strong activation on animals' faces. For the Horror classification, spatial features include people screaming and dark scenes. We also notice that text is an essential feature for the classification task, acting as a solid temporal marker in the sequence of tokens. Class activation maps are obtained via Grad-CAM[177] using the code provided by [71].

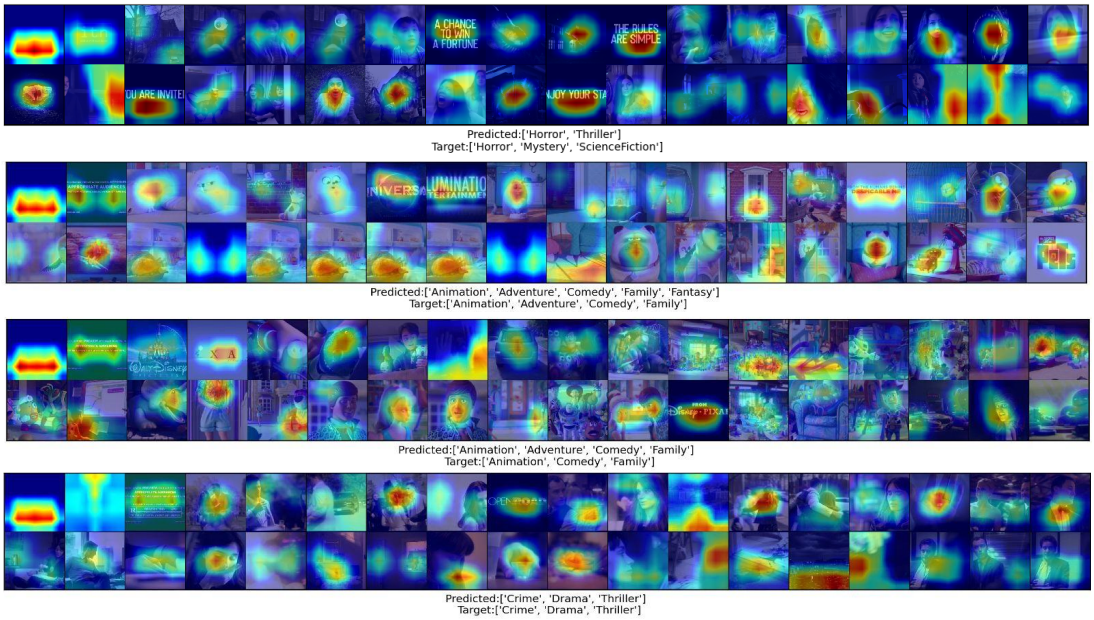



**Figure 4.3:** Class activation maps of randomly selected samples from the MMX-Trailer-20 test partition. Each sub-figure represents a series of input tokens from the spatial encoder $h$. The class activation maps show pixel regions in red to blue, with red regions representing high activation and, therefore, greater contribution to the predicted class labels.


**Figure 4.4:** Class activation maps for central scene frames from the MMX-Trailer-20 Dataset. A heatmap shows the class activation for the given genre. We observe that the genres are related to specific elements within the scenes such as objects and interactions.

In Fig 4.4, we show how the spatial CNN encoder identifies relevant features for input tokens. We find some surprising features are learned; for example, flowers strongly indicate a 'Romantic' token, while 'Action' tokens include architecture and clothing such as suits and ties. Comedy features include multiple people in the same shot.

We also show in Fig 4.5 that if we reduce the classification threshold following the sigmoid activation function in the final classification layer, we can obtain additional labels for coherent samples considering the content. In Fig 4.5, for example, the labels 'Action' and 'Drama' are appended to the classification label.

Finally, we also explore why the network incorrectly labels some trailers. In Fig 4.6, we find that the Comedy label has been incorrectly predicted as the trailer features a cat, and also that the sequence of scenes is interjected with frames of text which,h is a strong indicator of the temporal content of comedy trailers.

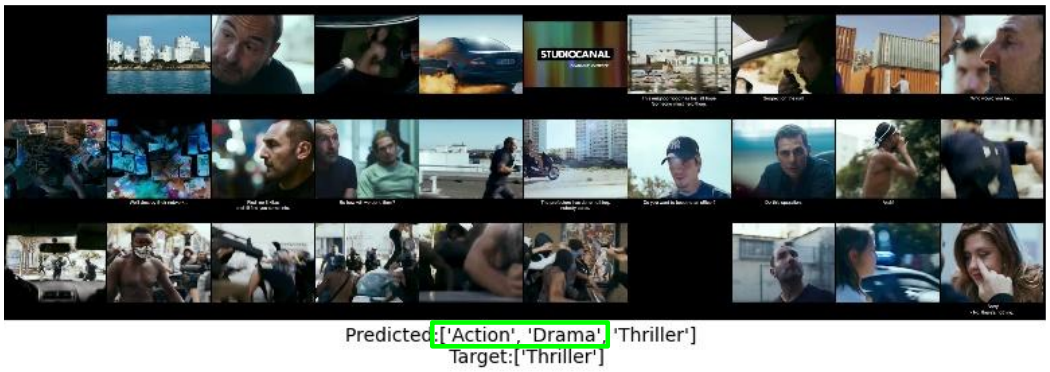



**Figure 4.5:** By reducing the threshold for a valid classification label to 0.3, we obtain the additional labels 'Action' and 'Drama', which is logical considering the content. Similarly, if we require just one Genre label, we can increase the threshold to 0.5 to obtain just the 'Thriller' label. Each frame shown represents 12 frames in the temporal stream and 1 in the spatial stream.


**Figure 4.6:** We explore why the incorrect label 'Comedy' has been predicted for this movie trailer. Animals and text are strong indicators of a 'Comedy' genre label. We also identify that the interjection of text frames is a strong temporal clue that the movie trailer has the 'Comedy' label.

## 4.4 Conclusion

In this chapter, we introduced a data and memory-efficient spatio-temporal attention network tailored to classify videos using spatio-temporal fusion. This network harnesses the advantages of convolutional inductive bias alongside the computational benefits of transformer networks, allowing for efficient video classification. Utilising static images and temporal context convolutional tokens, we developed an architecture that does data efficiently and supports several specific video understanding tasks. As discussed in the introduction, data and training efficiency are significant barriers to implementing video understanding systems for organisations with limited compute or data. This method effectively solves the problems, providing a methodology to learn from long videos in data and memory-restricted environments.

Despite its efficacy in classifying and recognising actions in long videos, our method encounters limitations in precisely locating when specific actions occur. Pinpointing these moments is crucial for advanced video understanding tasks such as video captioning and fine-grained retrieval, which require recognition and accurate temporal activity localisation of actions. To bridge this gap, the subsequent chapters will delve into two innovative methods aimed at

enhancing the capabilities of existing approaches to Temporal Activity Localisation. These approaches will demonstrate audio and text data integration to develop multimodal systems, providing a more comprehensive solution to accurately identifying and locating actions within video content.

# Chapter 5

# Multi-Resolution Audio-Visual Feature Fusion for Temporal Action Localization

In the previous chapters, we explored multimodal fusion and spatio-temporal understanding. However, many video understanding tasks also require temporal localisation. Temporal Action Localisation (TAL) detects the onset and offset of actions and their class labels in untrimmed and unconstrained videos. TAL is essential to many video understanding applications, allowing for in-video action retrieval, fine-grained video understanding, and action video captioning. As discussed in the literature review, two key challenges limit the practicability of using current TAL network architectures in real-world applications. The first is effectively leveraging multi-modalities, such as audio, to improve classification and localisation in a one-stage training setup. The other is ensuring that temporal invariance within these networks is robust and generalisable. Recently, the combined use of transformer networks and Feature Pyramid Networks (FPN) [237, 250, 40, 223, 180] has led to a significant boost in the performance and efficiency of TAL tasks by leveraging multi-resolution visual features. Using FPNs ensures that these networks feature more robust temporal invariance so that actions that may occur over different speeds and durations are identified correctly regardless of the speed in the video. However, there has yet to be a study on combining audio information in such network architectures for this task, specifically how to fuse audio information over different temporal resolutions. The challenge lies

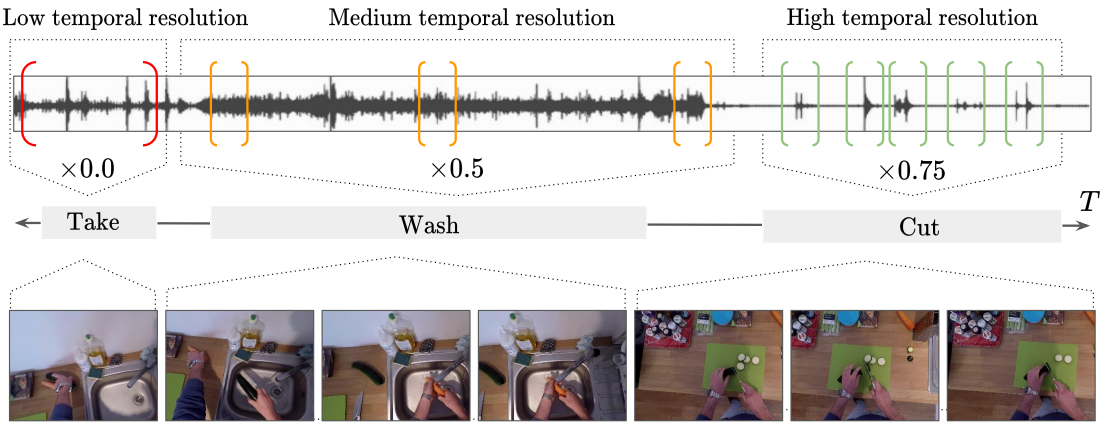



**Figure 5.1:** We use a Feature Pyramid Network (FPN) to encode audio-visual action features along different temporal resolutions. We then gate the fusion of the audio features depending on their application to the action classification and regression boundaries. For example, the action 'take' requires no audio, which is gated out. In contrast, the action 'cut' can be better localised by combining high-temporal resolution audio features with visual features. Our method learns both the temporal resolution and the gating values end-to-end.

in integrating audio and visual data and determining the density of audio information required across different FPN channels for various actions. While some channels require richer audio input to accurately identify action segments due to higher visual downsampling, others with more detailed visual cues might need less audio assistance. For instance, as shown in Fig 5.1, an action such as 'cut' can be better located using high-resolution (i.e. less downsampled) audio features. In contrast, an activity such as 'washing up' may only require some low-resolution audio information. A final example could be for an action such as 'pick-up', which requires no audio input. With this in mind, a fusion method for audio TAL should accommodate multiple temporal audio resolutions while also including a mechanism to gate audio information in specific temporal pathways.

This chapter presents a novel framework for Multi-Resolution Audio-Visual Feature Fusion (MRAV-FF) as the first step in solving the audio-visual temporal activity localisation issue. Our methodology is rooted in a hierarchical gated cross-attention fusion mechanism that adaptively combines audio and visual features over varying temporal scales. Unlike existing techniques, MRAV-FF weighs the significance of each modality's features at various temporal scales to improve the regression boundaries and classification confidence. Furthermore, our method can

be easily plugged into any FPN TAL architecture to boost performance when audio information is available.

## 5.1 Methodology

In this section, we formulate the problem definition of Temporal Action Localization (TAL) and provide details of the proposed method.

**Problem Definition**

Consider an untrimmed input video denoted as $\mathbb{V}$. The goal is to represent $\mathbb{V}$ as a set of feature vectors symbolized as $\mathbb{V} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}$. Each $\mathbf{x_t}$ corresponds to discrete time steps, $t = \{1, 2, \ldots, T\}$. Notably, the total duration $T$ is not constant and may differ across videos. For illustrative purposes, $\mathbf{x_t}$ can be envisaged as a feature vector extracted from a 3D convolutional network at a specific time $t$ within the video. The primary objective of TAL is to identify and label action instances in the input video sequence $\mathbb{V}$. These instances are collectively denoted as $\mathbb{Y} = \{y_1, y_2, \ldots, y_n\}$, where $N$ signifies the total number of action instances in a given video. This value can be variable across different videos. Each action instance, $y_i$, is defined by the tuple $y_i = (s_i, e_i, a_i)$, where $s_i$ represents the starting time or onset of the action instance, $e_i$ denotes the ending time or offset of the action instance, and $a_i$ specifies the action category or label.

The parameters must adhere to the conditions: $s_i, e_i \in \{1, \ldots, T\}$, $a_i \in \{1, \ldots, C\}$ (with $C$ indicating the total number of predefined action categories), and $s_i < e_i$, which ensures the starting time precedes the ending time for every action instance. Furthermore, alongside the visual feature set $\mathbb{X}$, we introduce an audio feature set $\mathbb{A}$. This set can be represented as $\mathbb{A} = \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_{t_{\text{audio}}}}\}$, spanning up to $T_{\text{audio}}$ time steps. Notably, the total duration $T_{\text{audio}}$ may or may not align with $T$ from the visual features, depending on the extraction mechanism and granularity of the audio features.

A significant challenge in multimodal TAL is devising an optimum method for fusing visual and audio features. This fusion aims to leverage complementary information from both modalities, enhancing the robustness and accuracy of action localisation and classification.

**Method Overview**

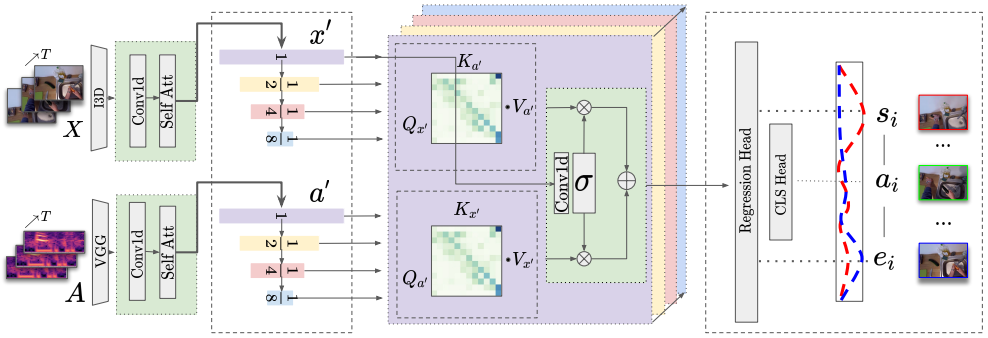



**Figure 5.2:** A high-level representation of our multi-resolution audio-fusion method. (a) We apply independent 1D convolutional filtering on audio and visual features to obtain a shared feature dimension. (b) Max-Pooling is applied to downsample each of the features. (c) After each downsampling operation we apply multi-headed cross attention for each temporal layer between audio and visual features. (d) The video features are then used as context to scale audio and visual attended embeddings. (e) The concatenated embedding is then used for both regression and classification.

As depicted in Fig 5.2, our proposed method is structured around three core components. First, video and audio features are extracted from untrimmed videos using frozen, pre-trained encoders. These encoders provide a robust foundation for capturing the inherent characteristics of the media without additional training overhead. Post-extraction, these features are further refined via a shallow convolution layer. Subsequently, they are channelled into a feature pyramid network where we apply cross-attention between audio and visual features at each temporal level. This mechanism ensures effective alignment and integration of features from diverse modalities and resolutions, facilitating the capture of complex temporal relationships. Finally, upon feature fusion, each temporal feature vector is processed by two dedicated decoders: one for regression, predicting action onsets and offsets, and the other for classification, identifying specific action class labels. This dual-decoder approach ensures accurate temporal localisation and semantic identification of each detected action.

### 5.1.1 Audio-Visual Temporal Fusion

Fusing audio and visual embeddings extracted from a video sequence is essential to enhancing the model's capability to interpret and analyse multimodal data. For each time step, audio embeddings are represented as $\mathbb{A} = \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_{t_{\text{audio}}}}\}$, and visual embeddings as $\mathbb{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}$.

The fusion process begins with the downsampling of these embeddings to reduce their dimensionality and highlight the most salient features. This reduction is accomplished through a max-pooling operation applied independently to each modality:

$$\mathbb{F}' = \text{MaxPool}(\mathbb{F}, \text{stride} = 2) \tag{5.1}$$

This operation effectively halves the temporal resolution of the feature sets, decreasing the computational demands for subsequent steps and concentrating the model's attention on dominant features.

Next, we implement a cross-attention mechanism to integrate information from the audio and visual domains efficiently. This process is pivotal for aligning and accentuating pertinent features across modalities. The cross-attention for any downsampled feature set $\mathbb{F}'$ is computed as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.2}$$

Where $Q = \mathbf{W_Q}\mathbb{F}'$, $K = \mathbf{W_K}\mathbb{F}'$, and $V = \mathbf{W_V}\mathbb{F}'$ are the query, key, and value matrices, respectively, with learnable parameters $\mathbf{W_Q}$, $\mathbf{W_K}$, and $\mathbf{W_V}$. The dimension $d_k$ is the scaling factor, typically set to the dimension of the key vectors.

Cross-modal interactions are facilitated through the computation of projection matrices for both audio and video acting as queries:

$$\mathbf{P_X} = \text{Attention}(\mathbf{x}'\mathbf{W_Q^X}, \mathbf{a}'\mathbf{W_K^A}, \mathbf{a}'\mathbf{W_V^A}) \tag{5.3}$$

$$\mathbf{P_A} = \text{Attention}(\mathbf{a}'\mathbf{W_Q^A}, \mathbf{x}'\mathbf{W_K^X}, \mathbf{x}'\mathbf{W_V^X}) \tag{5.4}$$

In these equations, $\mathbf{W}_{\mathbf{Q}}^{\mathbf{X}}$, $\mathbf{W}_{\mathbf{K}}^{\mathbf{X}}$, $\mathbf{W}_{\mathbf{V}}^{\mathbf{X}}$ are the query, key, and value matrices for the video features, and $\mathbf{W}_{\mathbf{Q}}^{\mathbf{A}}$, $\mathbf{W}_{\mathbf{K}}^{\mathbf{A}}$, $\mathbf{W}_{\mathbf{V}}^{\mathbf{A}}$ are those for the audio features. These matrices enable the mapping of audio features in response to video queries and vice versa. The resulting $\mathbf{P}_{\mathbf{X}}$ and $\mathbf{P}_{\mathbf{A}}$ represent the enriched feature sets, now blended with both audio and visual information, ready for advanced processing and decision-making.

### 5.1.2 Gated Audio-Visual Fusion

To further refine our fusion process, we introduce a gating mechanism that adaptively scales the contribution of audio and visual features based on the context of the visual content. For each downsampled visual feature $\mathbf{x}'$, a gating scalar $g$ is computed using a sigmoid function:

$$g = \sigma(\text{FC}(\mathbf{x}')) \tag{5.5}$$

Here, $\sigma$ is the sigmoid activation function, which ensures $g$ remains in the range $[0, 1]$, and FC represents a fully connected layer. The gating scalar adjusts the cross-modal projections as follows:

$$\mathbf{P}_{\mathbf{X},\text{GATED}} = g \cdot \mathbf{P}_{\mathbf{X}}, \tag{5.6}$$

$$\mathbf{P}_{\mathbf{A},\text{GATED}} = (1 - g) \cdot \mathbf{P}_{\mathbf{A}} \tag{5.7}$$

The aggregated feature representation after gating is then:

$$\mathbb{F}_{\text{GATED\_COMBINED}} = \text{Conv1D}([\mathbf{P}_{\mathbf{X},\text{GATED}}; \mathbf{P}_{\mathbf{A},\text{GATED}}]) \tag{5.8}$$

### 5.1.3 Regression and Classification

Each temporal layer outputs gated features to the classification and regression heads for detecting action instances. The output for each instant $t$ in feature pyramid layer $l$ is expressed as:

$$\hat{\mathbf{o}}_{\mathbf{t}}^{\mathbf{l}} = (\hat{\mathbf{c}}_{\mathbf{t}}^{\mathbf{l}}, \hat{\mathbf{d}}_{\mathbf{st}}^{\mathbf{l}}, \hat{\mathbf{d}}_{\mathbf{et}}^{\mathbf{l}}) \tag{5.9}$$

We adopt the same loss function as described in [196, 252, 250]. The total loss $\mathcal{L}$ is computed as:

$$\mathcal{L} = \mathcal{L}_{pos} + \mathcal{L}_{neg} \tag{5.10}$$

where $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ represent the losses for positive and negative samples, respectively. $\mathcal{L}_{pos}$ is calculated for the positive samples as follows:

$$\mathcal{L}_{pos} = \frac{1}{n_{pos}} \sum_{l,t} \mathbb{1}_{\{\mathbf{c}_t^l > 0\}} (\sigma_{IoU} \mathcal{L}_{cls} + \mathcal{L}_{reg}) \tag{5.11}$$

Here, $n_{pos}$ denotes the number of positive samples, and the indicator function $\mathbb{1}_{\{\mathbf{c}_t^l > 0\}}$ selects those samples. The term $\sigma_{IoU}$ represents the temporal IoU between the predicted segment and the ground truth action instance, which is used to re-weight the classification loss $\mathcal{L}_{cls}$ based on the quality of the regression. The regression loss is denoted as $\mathcal{L}_{reg}$.

$\mathcal{L}_{neg}$ is computed for the negative samples:

$$\mathcal{L}_{neg} = \frac{1}{n_{neg}} \sum_{l,t} \mathbb{1}_{\{\mathbf{c}_t^l = 0\}} \mathcal{L}_{cls} \tag{5.12}$$

In this case, $n_{neg}$ represents the number of negative samples, and the indicator function $\mathbb{1}_{\{\mathbf{c}_t^l = 0\}}$ identifies the samples where the predicted class is zero. Only the classification loss $\mathcal{L}_{cls}$ is applied here.

The total loss $\mathcal{L}$ combines the contributions from both positive and negative samples. The temporal IoU $\sigma_{IoU}$ is crucial in weighting the classification loss, ensuring that instances with better regression quality have a greater impact during training.

## 5.2 Implementation Details

This section offers additional details of the visual and audio feature extraction process.

### 5.2.1 Visual Features

We use the features provided by existing works in TAL [250, 124, 233]. For EPIC-Kitchens, features are extracted using a SlowFast network [62] pre-trained on EPIC-Kitchens [45]. During extraction, we use a 32-frame input sequence with a stride of 16 to generate a set of 2304-D features. For THUMOS 14, we use features extracted via a pre-trained I3D network [31] trained on Kinetics 400 [268], generating a 32-frame input sequence with a stride of 16 and 2048-D features. These are the same features used in all comparative works.

### 5.2.2 Audio Features

For the audio pre-processing and feature extraction, we followed a series of well-established steps to derive meaningful representations:

1. **Resampling:** All audio data was resampled to a uniform rate of 16 kHz in mono.

2. **Spectrogram Computation:** We computed the spectrogram by extracting magnitudes from the Short-Time Fourier Transform (STFT). This utilised a window size of 25 ms, a hop size of 10 ms, and a periodic Hann window for the analysis.

3. **Mel Spectrogram Mapping:** The computed spectrogram was then mapped to a mel scale, producing a mel spectrogram with 64 mel bins that cover the frequency range from 125 Hz to 7500 Hz.

4. **Log Mel Spectrogram Stabilization:** To enhance the stability and avoid issues with the logarithm function, we calculated a stabilised log mel spectrogram as:

$$\text{Log-Mel} = \log(\text{Mel-Spectrogram} + 0.01)$$

   Here, the offset of 0.01 prevents the computation of the logarithm of zero.

5. **Framing:** Finally, the derived features were segmented into non-overlapping examples spanning 0.96 seconds each. Every example encapsulates 64 mel bands and 96-time frames, with each frame lasting 10 ms.

Following extraction, the features are projected to 128-D features via a VGG audio encoder network [84] pre-trained on AudioSet [70]. The network outputs embeddings of shape $T \times 128$ where $T$ is the temporal input dimension.

## 5.3 Results

We evaluate our method on the Epic-Kitchens-100 and THUMOS-14 datasets. These are standard TAL datasets featuring detailed annotations for action localisation.

### 5.3.1 Datasets

#### EPIC-Kitchens 100 [44]

EPIC-Kitchens-100 is an egocentric dataset with two egocentric tasks: noun localisation (e.g. door) and verb localisation (e.g. open the door). It has 495 and 138 videos, with 67,217 and 9,668 action instances for training and inference, respectively. The number of action classes for nouns and verbs is 300 and 97. We follow all other methods [124, 250, 40, 249, 193], and report the mean average precision (mAP) at different intersections over union (IoU) thresholds with the average mAP computed over [0.1:0.5:0.1] in Table 5.1.

#### THUMOS-14 [101]

THUMOS-14 is a benchmark action detection dataset containing untrimmed videos across 20 action classes. It includes 200 validation videos and 213 test videos. The training set is sourced from the UCF101 dataset. For evaluation, we use the 213 test videos containing 3,358 action instances. Consistent with previous works [124, 250, 40, 249, 193], we report the mean average precision (mAP) at the different intersections over union (IoU) thresholds, specifically at [0.3, 0.4, 0.5, 0.6, 0.7]. The average mAP is computed and presented in Table 6.1.

### 5.3.2 Evaluation

#### EPIC-Kitchens 100

| Task | Method | tIoU | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | Avg |
| Verb | BMN [124, 45] | 10.8 | 9.8 | 8.4 | 7.1 | 5.6 | 8.4 |
| | G-TAD [233] | 12.1 | 11.0 | 9.4 | 8.1 | 6.5 | 9.4 |
| | ActionFormer [250] | 26.6 | 25.4 | 24.2 | 22.3 | 19.1 | 23.5 |
| | TemporalMaxer [193] | 27.8 | 26.6 | 25.3 | 23.1 | 19.9 | 24.5 |
| | ActionFormer + MRAV-FF | 27.6 | 26.8 | 25.3 | 23.4 | 19.8 | 24.6 |
| | TemporalMaxer + MRAV-FF | **28.5** | **27.4** | **26.0** | **23.7** | **20.12** | **25.1** |
| Noun | BMN [124, 45] | 10.3 | 8.3 | 6.2 | 4.5 | 3.4 | 6.5 |
| | G-TAD [233] | 11.0 | 10.0 | 8.6 | 7.0 | 5.4 | 8.4 |
| | ActionFormer [250] | 25.2 | 24.1 | 22.7 | 20.5 | 17.0 | 21.9 |
| | TemporalMaxer [193] | 26.3 | 25.2 | 23.5 | 21.3 | 17.6 | 22.8 |
| | ActionFormer + MRAV-FF | 26.4 | 25.4 | 23.6 | 21.2 | 17.4 | 22.8 |
| | TemporalMaxer + MRAV-FF | **27.4** | **26.2** | **24.4** | **21.8** | **17.9** | **23.5** |

**Table 5.1:** The performance of our proposed method on the EPIC-Kitchens 100 dataset. [45]

| Task | Method | tIoU | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | Avg |
| Verb | Concatenation | 28.02 | 26.96 | 25.5 | 23.48 | 19.87 | 23.89 |
| | Channel Pooling | 25.63 | 24.59 | 23.09 | 21.14 | 17.95 | 23.06 |
| | MRAV-FF | **28.5** | **27.4** | **26.0** | **23.7** | **20.12** | **25.1** |
| Noun | Concatenation | 26.39 | 25.42 | 23.57 | 21.19 | 17.42 | 22.8 |
| | Channel Pooling | 25.7 | 24.53 | 22.95 | 20.52 | 17.04 | 22.21 |
| | MRAV-FF | **27.4** | **26.2** | **24.4** | **21.8** | **17.9** | **23.5** |

**Table 5.2:** Results for an ablation experiment on EPIC-Kitchens 100 [45] TAL task, where we replace the MRAV-FF module with existing approaches to feature fusion, including concatenated projection and channel pooling. We observe that simple fusion methods hinder performance compared to uni-modal FPN networks, demonstrating the need for a more nuanced fusion strategy.

We show the effectiveness of our audio-fusion method in increasing the performance of unimodal models by adding our MRAV-FF to the best-performing existing FPN networks and evaluating them on EPIC-Kitchens 100. We also show how our method improves the performance of both ActionFormer and TemporalMaxer by +0.9 mAP and +0.4 mAP for verbs and +0.9 and +0.7 for nouns.

Furthermore, in Tab 5.3, we evaluate our method with other approaches to audio-visual fusion for TAL on EPIC-Kitchens. We show a significant increase in performance, which can be attributed to both the effectiveness of the FPN structure for audio-visual temporal pooling and our MRAV-FF fusion module. The lack of available comparative methods for audio-visual fusion further illustrates the importance of updated baselines in this field.

| Task | Method | tIoU | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | Avg |
| Verb | Damen [46] | 10.83 | 9.84 | 8.43 | 7.11 | 5.58 | 8.36 |
| | AGT [152] | 12.01 | 10.25 | 8.15 | 7.12 | 6.14 | 8.73 |
| | OWL [166] | 14.48 | 13.05 | 11.82 | 10.25 | 8.73 | 11.67 |
| | MRAV-FF | **28.5** | **27.4** | **26.0** | **23.7** | **20.12** | **25.1** |
| Noun | Damen [46] | 10.31 | 8.33 | 6.17 | 4.47 | 3.35 | 6.53 |
| | AGT [152] | 11.63 | 9.33 | 7.05 | 6.57 | 3.89 | 7.70 |
| | OWL [166] | 17.94 | 15.81 | 14.14 | 12.13 | 9.80 | 13.96 |
| | MRAV-FF | **27.4** | **26.2** | **24.4** | **21.8** | **17.9** | **23.5** |

**Table 5.3:** The performance of our proposed method on the EPIC-Kitchens 100 dataset [45] compared to existing approaches for audio-visual feature fusion on TAL. Our method demonstrates a large performance increase jointly attributed to adding feature pyramid architecture and our fusion strategy.

#### THUMOS 14

Finally, we also evaluate the method on the THUMOS14 dataset, which [97] contains 200 validation videos and 213 testing videos with 20 action classes. THUMOS14 presents a different challenge to egocentric videos since the videos are heavily edited and include many actions that need audio-visual alignment. For example, many videos are of sporting events with no localised audio information, contain music or narration, or have no audio. Due to these challenges, no existing TAL audio-visual fusion works, to our knowledge, test their methods on THUMOS14.

Following previous work [124, 125, 233, 256, 250], we trained the model on the provided validation set and evaluated it on the test set. This is because temporal boundaries are not available for the training samples. Our results in Tab 5.4 demonstrate that our method struggles to handle this audio-visual disparity, only improving on the $0.7$ IOU threshold. Our results are very similar to [193], suggesting that the audio provides no additional helpful information for the localisation task. Reviewing the Thumos dataset, we can observe that many of the classes are sports-related or instructional, with commentary, music and narration. To deal with such video data, our method could be improved by adding a speech-to-text network for narrated videos to look for additional context clues for boundary detection or to develop an audio-visual alignment classifier before processing the embeddings to the gating network to filter out such samples.

| Type | Model | Feature | tIoU↑ | | | | time(ms) ↓ |
|---|---|---|---|---|---|---|---|
| | | | 0.3 | 0.5 | 0.7 | Avg. | |
| Two-Stage | BMN [124] | TSN [213] | 56.0 | 38.8 | 20.5 | 38.5 | 483* |
| | DBG [121] | TSN [213] | 57.8 | 39.8 | 21.7 | 39.8 | — |
| | G-TAD [233] | TSN [213] | 54.5 | 40.3 | 23.4 | 39.3 | 4440* |
| | BC-GNN [14] | TSN [213] | 57.1 | 40.4 | 23.1 | 40.2 | — |
| | TAL-MR [256] | I3D [31] | 53.9 | 45.4 | 28.5 | 43.3 | >644* |
| | P-GCN [249] | I3D [31] | 63.6 | 49.1 | — | — | 7298* |
| | P-GCN [249] +TSP [6] | R(2+1)1 D [203] | 69.1 | 53.5 | 26.0 | 50.5 | — |
| | TSA-Net [74] | P3D [162] | 61.2 | 46.9 | 25.2 | 45.1 | — |
| | MUSES [130] | I3D [31] | 68.9 | 56.9 | 31.0 | 53.4 | 2101* |
| | TCANet [160] | TSN [213] | 60.6 | 44.6 | 26.7 | 44.3 | — |
| | BMN-CSA [186] | TSN [213] | 64.4 | 49.2 | 27.8 | 47.7 | — |
| | ContextLoc [266] | I3D [31] | 68.3 | 54.3 | 26.2 | 50.9 | — |
| | VSGN [255] | TSN [213] | 66.7 | 52.4 | 30.4 | 50.2 | — |
| | RTD-Net [192] | I3D [31] | 68.3 | 51.9 | 23.7 | 49.0 | >211* |
| | Disentangle [267] | I3D [31] | 72.1 | 57.0 | 28.5 | 53.5 | — |
| | SAC [239] | I3D [31] | 69.3 | 57.6 | 31.5 | 54.0 | — |
| Single-Stage | A²Net [240] | I3D [31] | 58.6 | 45.5 | 17.2 | 41.6 | 1554* |
| | GTAN [134] | P3D [162] | 57.8 | 38.8 | — | — | — |
| | PBRNet [128] | I3D [31] | 58.5 | 51.3 | 29.5 | — | — |
| | AFSD [122] | I3D [31] | 67.3 | 55.5 | 31.1 | 52.0 | 3245* |
| | TAGS [148] | I3D [31] | 68.6 | 57.0 | 31.8 | 52.8 | — |
| | HTNet [106] | I3D [31] | 71.2 | 61.5 | 39.3 | 58.0 | — |
| | TadTR [131] | I3D [31] | 74.8 | 60.1 | 32.8 | 56.7 | 195* |
| | GLFormer [82] | I3D [31] | 75.9 | 67.2 | 41.8 | 62.9 | — |
| | AMNet [131] | I3D [31] | 76.7 | 66.8 | 42.7 | 63.3 | — |
| | ActionFormer [250] | I3D [31] | 82.1 | 71.0 | 43.9 | 66.8 | 80 |
| | ActionFormer [250] + GAP [147] | I3D [31] | 82.3 | 71.4 | 44.2 | 66.9 | >80 |
| | TemporalMaxer | I3D [31] | **82.8** | **71.8** | 44.7 | **67.7** | **50** |
| | TemporalMaxer + MRAVFF | I3D [31] + Audio [84] | 82.2 | 71.5 | **45.3** | 67.4 | 60 |

**Table 5.4:** Performance of our method on the THUMOS dataset for TAL. We observe that audio-visual fusion on edited videos is much more challenging than the raw-video setting due to the addition of background music, narration, and audio-visual misalignment.

### 5.3.3 Ablation Experiments

We perform initial ablation experiments to evaluate the performance of our proposed method and present the results in Tab 5.2. Each experiment is conducted on EPIC-Kitchens, where we edit the temporal fusion method in each temporal block. We first exchange our MRAV-FF temporal block for simple feature fusion in which we concatenate and project the audio-visual features at each temporal scale via a 1D-CNN. We notice that this harms network performance over unimodal features, demonstrating the need for a gated approach to fusion. Similarly, we replace the block with a max-pooling layer inspired by [193], which pools channel-wise for feature fusion. Again, this method hurts network performance. We note that although the average increase in the performance of our method appears small when compared with channel pooling over each temporal dimension, we improve performance significantly in the lower tIOU thresholds. As we increase the threshold, the task becomes much easier, and as such, other methods perform comparatively well.

## 5.4 Conclusion

This chapter discussed the challenge of fusing audio-visual features for more detailed video-understanding tasks such as temporal action localisation. We show how adding audio features does not improve performance and requires careful design considerations depending on the dataset and application. While we significantly improve performance on the EPIC-KITCHENS dataset, which contains long unedited videos, our method needs to improve on datasets where the audio is heavily edited or misaligned, for example, when there is background music or narration. Unfortunately, only a few datasets contain detailed action annotations with unedited audio channels. Obtaining these annotations can be time-consuming and expensive, so they are only feasible in some real-world settings. In the next chapter, we address this problem and introduce a new method to perform temporal action localisation with just a few labelled examples by leveraging pre-trained vision-language networks and prompt learning.

# Chapter 6

# Prompt Learning with Optimal Transport for Few-Shot Temporal Action Localization

In the previous chapter, we introduced a novel method for audio-visual fusion in temporal action localisation. Now, we focus on incorporating text as an additional modality, which presents unique challenges and opportunities. Vision-language models for video processing allow users to query videos using natural language, enabling applications such as video editing, retrieval, and action detection. This capability transforms fields like automated video summarisation, content recommendation, advanced security systems, and human-computer interaction.

For example, vision-language models can automate the search for specific scenes based on textual descriptions in video editing, significantly reducing manual editing time. In video retrieval, users can search for clips containing particular actions or events described by text, which is invaluable for media companies with vast video archives. Additionally, in action detection, these models can improve the precision and recall of identifying activities within videos, which is crucial for applications like surveillance and sports analytics.

Despite these promising applications, implementing vision-language models in real-world scenarios presents significant challenges. One primary difficulty is the scarcity of annotated datasets essential for training these models. Creating datasets that link text descriptions to correspond-

ing video segments is labour-intensive and time-consuming. The diversity and complexity of real-world videos, with varying lighting conditions, camera angles, and background activities, further complicate the annotation process. Another challenge is the high computational cost of training and deploying vision-language models, which often require substantial processing power and memory. Moreover, integrating temporal dynamics into these models—understanding how actions evolve within a video—adds another layer of complexity.

Few-shot learning emerges as a solution, aiming to generalise from a limited number of examples and reducing the need for large annotated datasets. Current approaches to few-shot learning for temporal action localisation typically involve meta-learning, where each test video is aligned with a small subset of training data in many 'episodes.' These methods require learning a model from initialisation, consuming significant memory and compute resources. Adapting pre-trained image encoders from large-scale vision-language pre-trained (VLP) models like CLIP [164] and Align [119] is one approach, but these networks are prone to over-fitting in few-shot scenarios. Moreover, adapting image CLIP encoders to video ignores the rich temporal dynamics essential for classification.

A recent training paradigm, prompt learning, reduces the number of trainable parameters by fixing all model parameters except for a learnable context vector added to the prompt to align it with image features. However, single prompt learning methods tend to optimise towards the average of the features, which lacks discriminative ability. In temporal action localisation, determining the exact boundaries of an action is crucial. A single learnable prompt will likely have high cosine similarity over a wide range of temporal features, leading to poor action boundary detection and, instead, using multiple prompts for each action, encompassing various views over the video, where each prompt aligns with different foreground or background views.

In this chapter, we introduce a novel methodology to overcome these challenges. We employ a pre-trained contrastive-language image network, which we adapt to map video features in the text captions. Rather than fine-tune the network, we introduce additional, parameterised context prompts for each caption and align these with the extracted features using Optimal Transport. As shown in Fig 6.1, This approach enhances the classification of action boundaries by utilising multiple context prompts for each class while requiring only a few trainable parameters. Furthermore, integrating a Feature Pyramid Network ensures these prompts are aligned across

multiple temporal scales and resolutions, improving the model's generalisability across diverse temporal speeds and contexts.

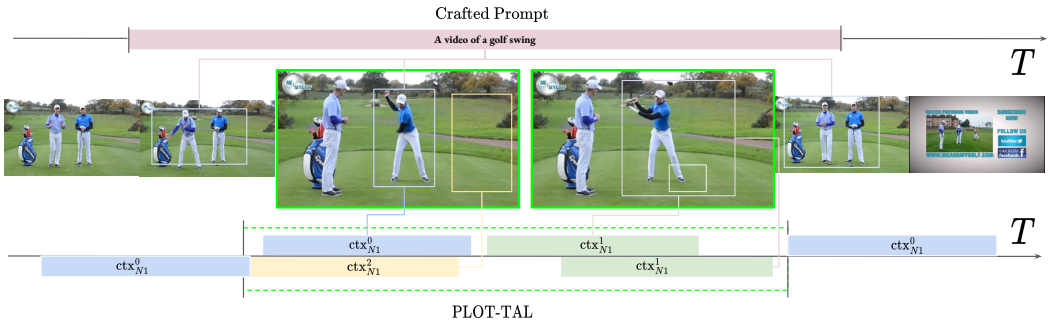



**Figure 6.1:** Existing methods either learn a single prompt to identify the location and class of a given action or construct a prompt, but learning multiple complimentary views can help with class generalisation and temporal discrimination within the video. Green frames indicate the ground truth foreground features. A single handcrafted prompt will have high cosine similarity over all of the video frames. However, learning multiple prompts enables us to learn specific views that can help discriminate between background and foreground features. In this example, the handcrafted prompt contains elements that will appear throughout the video, while the learned prompts align with individual elements that can be composed to identify the foreground feature.

Through this approach, we aim to significantly reduce the barriers to entry for TAL applications, making it more feasible to apply these technologies to a broader range of domains and datasets with limited annotations.

### 6.0.1 Optimal Transport

This chapter primarily focuses on integrating Optimal Transport (OT) to map text prompts onto specific foreground and background features in video content, facilitating more accurate temporal activity localisation. The foundational principle of OT, tracing back to the 18th-century work of Gaspard Monge [208], involves determining the most cost-effective method for relocating mass from one distribution to another. Over the years, OT has been applied to diverse fields such as economics [69], fluid dynamics [39], and more contemporarily, in machine learning and computer vision [197].

One of the core strengths of OT lies in its ability to handle distributional differences effectively. Unlike pointwise similarity measures, OT is designed to align entire distributions, making it

ideal for comparing the sequence of features extracted from video data with the distribution of learned prompts. This characteristic allows OT to account for the global structure of these distributions, leading to more robust and context-aware alignment.

Furthermore, OT excels in contextual matching. By considering the context within which features exist, OT facilitates more accurate alignment of sequences, which is crucial for actions defined by a series of motions. This is particularly beneficial compared to simpler metrics that may overlook broader temporal contexts, ensuring that the model captures the intricate relationships between different temporal segments.

Additionally, OT provides soft assignment and flexibility in matching. This capability allows OT to distribute the ”mass” across multiple prompts, accommodating scenarios where a video segment corresponds to multiple actions or where action boundaries are blurred. Such flexibility represents a significant improvement over traditional similarity-based methods that often enforce hard assignments, potentially leading to suboptimal matches.

Historically, the computational intensity and sensitivity to outliers of the OT problem posed significant challenges in direct applications, particularly within the high-dimensional settings typical in machine learning and computer vision domains. These obstacles were largely mitigated by the introduction of entropy regularisation [41], simplifying the computational process through iterative scaling algorithms and ensuring a unique, stable solution. Such advancements facilitate the use of parallel computing, substantially reducing the time required for computations [42, 24].

Recent studies have showcased novel applications of OT in machine learning. For instance, Yang et al. utilised OT for efficient attention allocation between optical flow and RGB features, aiming to learn a structure matrix that encapsulates dependencies among modalities within each frame [239]. Another notable application by Chen et al. involves aligning multiple text prompts with feature maps for enhanced few-shot image classification. They proposed treating local visual features and prompts as samplings from two discrete distributions. They also employed OT to foster fine-grained cross-modal integration, utilising CLIP multi-head self-attention layer outputs to extract feature maps [35].

Our work explores the applicability of similar OT strategies to temporal activity localisation tasks. By conceptualising the optimal transport between temporal features and multiple text prompts as samplings from discrete distributions, we aim to harness OT’s potential to refine the

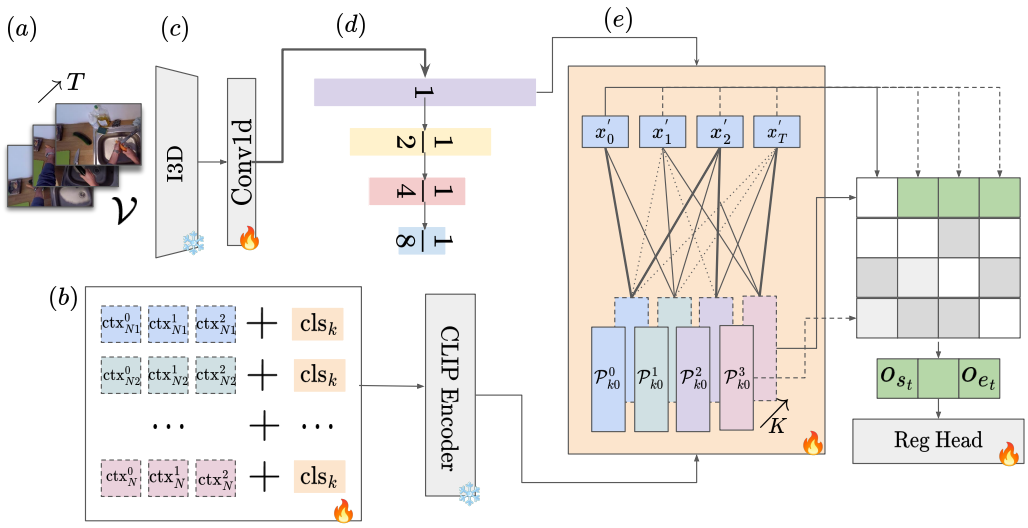



**Figure 6.2:** An overview of the approach. **A.)** We sample $T$ overlapping segments of videos $\mathcal{V}$. **B.)** For each class label $K$, we randomly initialize $N$ learnable vectors concatenated with the class label. **C.)** Video features are extracted via a pre-trained 3D CNN encoder (I3D ) while $N$ prompts for each class $k$ are also extracted via the pre-trained CLIP text encoder. **D.)** We temporally downsample the features using max-pooling. **E.)** We search the optimal transport plan between the $N$ prompt features and video segments at each temporal level. Following this stage, we sum all $N$ vectors for each $K$. **F.)** At each temporal level $L$, we compute the cosine similarity between each prompt vector $P_k$ and each video segment $x_i$ and then apply a threshold to retrieve action candidates. These candidates are passed to the regression head, minimising the distance between the start and end actions and each embedding. Only components with the fire symbol are trained, and all others are frozen.

alignment of textual and visual modalities, thereby improving model performance in dynamic video content contexts.

## 6.1 Methodology

We propose a novel framework for Temporal Action Localisation (TAL) in untrimmed videos. Our approach integrates pre-trained feature extraction, adaptive prompt learning, and efficient feature-prompt alignment via the Sinkhorn algorithm. An overview of the network architecture is shown in Fig 4.1

We aim to learn a generalisable representation for each action instance tuple, $(s_i, e_i, a_i)$, where $s_i$ represents the starting time or onset of the action instance, $e_i$ denotes the ending time or offset

of the action instance, and $a_i$ specifies the action category or label. We use minimal annotated examples, optimising for the accurate classification of action types and precise localisation of their temporal boundaries. Integrating pre-trained feature extractors minimises the model's dependency on extensive training data, aligning with the resource-intensive nature of video processing tasks.

Considering an untrimmed input video as $\mathbb{V}$, we represent it as a set of feature vectors tokenised as $\mathbb{V} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}$. Each $\mathbf{x_t}$ corresponds to discrete time steps, $t = \{1, 2, \ldots, T\}$. Notably, the total duration $T$ is not constant and may differ across videos. For illustrative purposes, $\mathbf{x_t}$ can be envisaged as a feature vector extracted from a 3D convolutional network at a specific time $t$ within the video. The primary objective of TAL is to identify and label action instances in the input video sequence $\mathbb{V}$. These instances are collectively denoted as $\mathbb{Y} = \{y_1, y_2, \ldots, y_n\}$, where $N$ signifies the total number of action instances in a given video. This value can be variable across different videos.

The parameters must adhere to the conditions: $s_i, e_i \in \{1, \ldots, T\}$, $a_i \in \{1, \ldots, C\}$ (with $C$ indicating the total number of predefined action categories), and $s_i < e_i$, which ensures the starting time precedes the ending time for every action instance.

In the few-shot setting, we aim to learn some general representation of each action instance $\mathbb{Y}$ using only a limited number of annotations that we can later classify the action onsets, offsets, and class labels of unseen videos. Since video understanding tasks are typically resource and data-intensive, we also want to minimise the number of trainable parameters in the model.

### 6.1.1 Feature Extraction and Representation

Given the untrimmed input video $\mathbb{V}$, we extract a sequence of feature vectors $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}$ corresponding to each time step $t$, using a 3D convolutional network. The extraction process is formalised as follows:

$$\mathbf{x_t} = \boldsymbol{f}_{\text{cnn}}(\mathbf{v_t}), \quad t = 1, 2, \ldots, T, \tag{6.1}$$

where $\mathbf{v_t}$ denotes the input from the video at time $t$, and $\boldsymbol{f}_{\text{cnn}}$ represents the 3D convolutional network function.

To further refine these features and incorporate contextual information, we apply a 1D convolutional layer:

$$\mathbf{x}'_{\mathbf{t}} = \boldsymbol{f}_{\text{conv}}(\mathbf{x}_{\mathbf{t}}), \quad t = 1, 2, \ldots, T, \tag{6.2}$$

where $\boldsymbol{f}_{\text{conv}}$ denotes the convolutional operation, which is designed to enhance the local temporal feature representation as demonstrated in [250, 181, 193].

### 6.1.2 Adaptive Prompt Learning

In the few-shot training set, we introduce additional learnable prompts for each class to ensure we can align with multiple views in various temporal dimensions. For each action category $k$, we generate $N$ prompts $\mathbb{P}_k = \{\mathbf{p_{k1}}, \mathbf{p_{k2}}, \ldots, \mathbf{p_{kn}}\}$, each consisting of the class label and $n_{\text{ctx}}$ context vectors, encoded as:

$$\mathbf{p_{ki}} = \boldsymbol{f}_{\text{clip}}(\text{label}_k, \text{ctx}_{k1}, \ldots, \text{ctx}_{kn_{\text{ctx}}}), \tag{6.3}$$

where $\boldsymbol{f}_{\text{clip}}$ signifies the encoding function from a pre-trained CLIP model, integrating the semantic content of the action categories into the model.

### 6.1.3 Optimal Transport with Sinkhorn Algorithm

We aim to align each class's $N$ learnable prompts with the most similar video features in cosine similarity. This is performed via optimal transport with the Sinkhorn Algorithm [41] to ensure the method is tractable and efficient.

To align the refined video features $\{\mathbf{x'_1}, \mathbf{x'_2}, \ldots, \mathbf{x'_t}\}$ with the adaptive prompts $\mathbb{P}_k$ for each action category $k$, we employ the Optimal Transport (OT) metric as a critical tool. The OT metric quantifies the discrepancies between two distributions, which, in our context, are the distributions of video features and prompt embeddings. Formally, let us denote the distributions of video features and prompts as:

$$\mathbf{u} = \sum_{t=1}^{T} u_t \delta_{\mathbf{x'_t}} \quad \text{and} \quad \mathbf{v_k} = \sum_{i=1}^{N} v_{ki} \delta_{\mathbf{p_{ki}}}, \tag{6.4}$$

where $\delta_{\mathbf{x}'_\mathbf{t}}$ and $\delta_{\mathbf{p_{ki}}}$ represent Dirac delta functions centered at the video feature $\mathbf{x}'_\mathbf{t}$ and prompt embedding $\mathbf{p_{ki}}$, respectively. The vectors $u$ and $v_k$ are normalized such that $\sum_{t=1}^{T} u_t = 1$ and $\sum_{i=1}^{N} v_{ki} = 1$, ensuring they represent discrete probability distributions.

The cost matrix $\mathbf{C}$, with elements $C_{ti}$, defines the cost of transporting mass from video feature $\mathbf{x}'_\mathbf{t}$ to prompt embedding $\mathbf{p_{ki}}$. The cost is typically inversely related to the similarity between $\mathbf{x}'_\mathbf{t}$ and $\mathbf{p_{ki}}$, such as $C_{ti} = 1 - \text{sim}(\mathbf{x}'_\mathbf{t}, \mathbf{p_{ki}})$. The goal of OT is to find a transport plan $\mathbf{T}$ that minimizes the total transport cost:

$$d_{\text{OT}}(\mathbf{u}, \mathbf{v_k}|\mathbf{C}) = \min_{\mathbf{T}} \langle \mathbf{T}, \mathbf{C} \rangle, \quad \text{subject to} \quad \mathbf{T}\mathbf{1}_N = u,\ \mathbf{T}^\top \mathbf{1}_M = v_k,\ \mathbf{T} \in \mathbb{R}_+^{M \times N}. \quad (6.5)$$

Due to the computational intensity of solving this problem, we apply the Sinkhorn Algorithm [41] with entropic regularisation for efficient optimisation. The regularization introduces an entropy term $h(\mathbf{T}) = -\sum_{t,i} T_{ti} \log T_{ti}$ to the optimization objective:

$$d_{\text{OT},\lambda}(\mathbf{u}, \mathbf{v_k}|\mathbf{C}) = \min_{\mathbf{T}} \langle \mathbf{T}, \mathbf{C} \rangle - \lambda h(\mathbf{T}), \quad \text{subject to} \quad \mathbf{T}\mathbf{1}_N = u,\ \mathbf{T}^\top \mathbf{1}_T = v_k. \quad (6.6)$$

The Sinkhorn algorithm iteratively adjusts $\mathbf{T}$ to satisfy the constraints efficiently, using updates based on matrix scaling operations. The iterative process converges to an optimal transport plan $\mathbf{T}^*$, representing the optimal alignment between video features and prompts. This alignment guides identifying and classifying action instances by optimally matching video segments to their corresponding semantic labels.

### 6.1.4 Temporal Pyramid and Feature Integration

Since actions can occur at a wide range of speeds and temporal intervals, as in the previous chapter, we utilise a temporal feature pyramid network [193] to optimise prompt alignments over multiple temporal scales. To do so, we construct a temporal pyramid from the refined features $\{\mathbf{x}'_\mathbf{1}, \mathbf{x}'_\mathbf{2}, \ldots, \mathbf{x}'_\mathbf{t}\}$ by applying a max-pooling operation at each level of the pyramid with a stride of 2, effectively halving the temporal dimension at each step:

$$\mathbb{X}'_l = \text{MaxPool}(\mathbb{X}'_{l-1}), \quad l = 2, \ldots, L, \quad (6.7)$$

where $\mathbb{X}'_l$ represents the set of features at level $l$ of the pyramid. This hierarchical structure is crucial for integrating temporal information across different scales, enabling the nuanced capture of action dynamics from coarse to fine temporal resolutions.

### 6.1.5 Multi-Resolution Temporal Alignment

For each level of the temporal pyramid, we employ the Optimal Transport (OT) metric to align the video features at that scale, $\{\mathbf{x'_{1,l}}, \mathbf{x'_{2,l}}, \ldots, \mathbf{x'_{t,l}}\}$, with the adaptive prompts $\mathbb{P}_k$ corresponding to each action category $k$. This alignment is performed separately at each level $l$ of the feature pyramid, allowing for a multi-scale analysis sensitive to the temporal granularity of actions.

The OT problem at each pyramid level $l$ is defined as follows:

$$d_{\text{OT},\lambda}(\mathbf{u_l}, \mathbf{v_{k,l}}|\mathbf{C_L}) = \min_{\mathbf{T_L}} \langle \mathbf{T_L}, \mathbf{C_L} \rangle - \lambda h(\mathbf{T_L}), \tag{6.8}$$

subject to $\mathbf{T_L}\mathbf{1}_N = u_l,\ {\mathbf{T_L}}^\top \mathbf{1}_T = v_{k,l}$, for each pyramid level $l$. Here, $\mathbf{C_L}$ represents the cost matrix at level $l$, and $\mathbf{u_l}, \mathbf{v_{k,l}}$ are the distributions of video features and prompts at this specific scale, respectively.

We then adopt the same two-stage optimisation process proposed in [35], which consists of an inner loop, during which we find the optimal alignment between features and prompts, and then an outer loop in which we update the learnable parameters.

Within the inner loop, for each level $l$ of the temporal pyramid, we fix the feature sets $\mathbb{F}_\mathbb{L}$ and prompt sets $\mathbb{G}_{\mathbb{K},\mathbb{L}}$, and minimize the OT distance to optimally align $\mathbb{G}_{\mathbb{K},\mathbb{L}}$ to $\mathbb{F}_\mathbb{L}$. The cost matrix $\mathbf{C_L}$ is computed to reflect the cosine similarity between the features and prompts at that scale:

$$\mathbf{C_L} = 1 - {\mathbb{F}_\mathbb{L}}^\top \mathbb{G}_{\mathbb{K},\mathbb{L}}. \tag{6.9}$$

This minimization results in an optimized transport plan $\mathbf{T^*_L}$ and the corresponding OT distance $d_{\text{OT},l}(k)$.

In the outer loop, with the transport plans $\mathbf{T^*_L}$ determined for each level of the pyramid, we update the learnable vectors across all scales. This holistic optimisation ensures that our model can

adaptively align video features with textual prompts at varying temporal resolutions, enhancing its ability to localise and classify actions within untrimmed videos accurately.

### 6.1.6 Decoder Architecture

Following the multi-scale alignment of video features with adaptive prompts through the Optimal Transport framework, our decoder architecture is designed to leverage these aligned features for sequence labeling and action boundary detection. Unlike conventional CNN-based decoders, our approach utilises the optimally aligned video features across each scale of the temporal pyramid to predict a sequence of action labels.

For each temporal scale $l$, the decoder generates a probability distribution for action classifications using a sigmoid activation function:

$$\mathbf{c_l} = \sigma(\mathbf{x'_{t,l}}), \tag{6.10}$$

where $\mathbf{x'_{t,l}}$ denotes the aligned video feature at time $t$ and scale $l$.

Furthermore, to accurately predict the start and end times of actions, a lightweight regression mechanism is employed:

$$\mathbf{o_l} = \text{ReLU}(\mathbf{W}_o \cdot \mathbf{x'_{t,l}}), \tag{6.11}$$

where $\mathbf{W}_o$ represents trainable weights for the regression task.

By integrating the optimally aligned features from multiple scales of the temporal pyramid, the decoder architecture effectively enhances the model's capability to recognise and localise actions, considering the diverse temporal scales inherent in video data.

### 6.1.7 Learning Objective

The learning objective aims to minimise the total loss, encompassing TAL's classification and localisation aspects. This is achieved through a combination of Focal Loss for handling class imbalance in action classification and DIoU Loss for improving the accuracy of action boundary predictions:

$$\mathcal{L}_{\text{total}} = \sum_{t=1}^{T} \left( \mathcal{L}_{\text{cls}}(\hat{c}_t, c_t) + \mathbb{1}_{\{c_t>0\}} \mathcal{L}_{\text{reg}}(\hat{o}_{s_t}, \hat{o}_{e_t}, o_{s_t}, o_{e_t}) \right) \tag{6.12}$$

where $\mathcal{L}_{\text{cls}}$ is the classification loss computed using Focal Loss as outlined in the previous chapter, $\mathcal{L}_{\text{reg}}$ is the regression loss computed using DIoU Loss, $\hat{c}_t$ and $c_t$ represent the predicted and true action categories, respectively, and $\hat{o}_{s_t}, \hat{o}_{e_t}, o_{s_t}, o_{e_t}$ denote the predicted and true start and end times of actions. The indicator function $\mathbb{1}_{\{c_t>0\}}$ ensures that regression loss is only applied to positive samples, i.e., time steps where an action is present.

## 6.2 Implementation Details

This section provides details on implementation, including feature extraction and training. We also give algorithms to demonstrate the two-stage prompt alignment method.

### 6.2.1 Feature Extraction

Features are extracted from a pre-trained I3D network [268] trained on the Kinetics-600 dataset [109, 268] in a supervised setting. We extract the optical flow and RGB output embeddings, which are then concatenated to form a $2048 \times T$ embedding, where $T$ is the total number of video segments. Each video segment refers to 16 frames sampled at 30 FPS with a stride of 4 frames. This is the standard feature extraction pipeline used in all previous TAL works [250, 181]. To deal with variable frame lengths $T$, we pad all samples to $T = 2048$, which accounts for the length of all videos. During training, we include a mask to represent the zero-padded regions and apply the mask after each operation.

### 6.2.2 Training

We train each model for 100 epochs, except for when we increase the number of shots above 15, in which case we train for 200. We randomly initialise the $ctx$ embedding vectors and append them to the start of the prompt. All models are trained with a batch size of 2 on a single NVIDIA RTX 3090 24GB GPU. The memory required for training the model on THUMOS'14 with a batch size of 2 and when $N = 4$ is 5GB. We include a summary of the method in 1.

The optimal transport is optimised in a two-stage process as proposed in [35] where we find the transport cost between the video features and prompts in the inner loop. After converging

**Algorithm 1** Overview of TAL-PLOT method

**Input:** Untrimmed input video $\mathbb{V}$
**Output:** Action instances $\mathbb{Y} = \{y_1, y_2, \ldots, y_n\}$
**Feature Extraction and Representation:**
**for** $t = 1$ to $T$ **do**
    Extract feature vector $\mathbf{x_t} = \boldsymbol{f}_{\text{cnn}}(\mathbf{v_t})$ using a 3D CNN
    Refine features $\mathbf{x'_t} = \boldsymbol{f}_{\text{conv}}(\mathbf{x_t})$ with a 1D convolutional layer
**end for**
**Adaptive Prompt Learning:**
**for** each action category $k$ **do**
    Generate $N$ prompts $\mathbb{P}_k = \{\mathbf{p_{k1}}, \mathbf{p_{k2}}, \ldots, \mathbf{p_{kn}}\}$ using $\boldsymbol{f}_{\text{clip}}$
**end for**
**Optimal Transport with Sinkhorn Algorithm:**
**for** each action category $k$ **do**
    Align features $\{\mathbf{x'_1}, \ldots, \mathbf{x'_t}\}$ with prompts $\mathbb{P}_k$ using OT
**end for**
**Temporal Pyramid and Feature Integration:**
Construct temporal feature pyramid $\mathbb{X}'_l$ with max-pooling
**Multi-Resolution Temporal Alignment:**
**for** $l = 1$ to $L$ **do**
    Align features at level $l$ of the pyramid with $\mathbb{P}_k$
**end for**
**Decoder Architecture:**
Use aligned features to predict action labels $\gtrless$ and boundaries $\mathbb{O}_{\mathbb{L}}$
**Learning Objective:**
Minimize total loss $\mathcal{L}_{\text{total}}$ with Focal Loss and DIoU Loss
**return** $\mathbb{Y}$

the Sinkhorn algorithm, we use the backward pass to update the learnable prompts. For the parameters, we follow the setup in [35] where $\delta = 0.01$, $\lambda = 0.1$, and we perform 100 iterations within the inner loop. We generate results over 4 random seeds and report the average. Further details are provided in Algorithm 2.

**Algorithm 2** Optimal Transport Sinkhorn Algorithm for Few-Shot TAL

**Input:** Untrimmed input video $\mathbb{V}$, pre-trained model features $\boldsymbol{f}_{\text{cnn}}$, number of prompts $N$, entropy parameter $\lambda$, maximum number of iterations $T_{\text{in}}$, $T_{\text{out}}$
**Output:** Optimized prompt parameters $\{\omega_n\}_{n=1}^{N}$
Initialize prompt parameters $\{\omega_n\}_{n=1}^{N}$
**for** $t_{\text{out}} = 1$ to $T_{\text{out}}$ **do**
  Obtain a visual feature set $\mathbb{F} \in \mathbb{R}^{M \times C}$ with the visual encoder $\boldsymbol{f}_{\text{cnn}}(\mathbf{x_t})$
  Generate prompt feature set $\mathbb{G}_k \in \mathbb{R}^{N \times C}$ for each class with textual encoder $\boldsymbol{g}(\text{label}_k, \text{ctx}_{k1}, \ldots, \text{ctx}_{kn_{\text{ctx}}})$
  Calculate the cost matrix $\mathbf{C}_k = 1 - \mathbb{F}^{\top}\mathbb{G}_k$ for each class
  Calculate the OT distance with an inner loop:
  Initialize $v^{(0)} = 1, \delta = 0.1, \Delta v = \infty$
  **for** $t_{\text{in}} = 1$ to $T_{\text{in}}$ **do**
    Update $u^{(t_{\text{in}})} = u/(\exp(-\mathbf{C}/\lambda)v^{(t_{\text{in}}-1)})$
    Update $v^{(t_{\text{in}})} = v/(\exp(-\mathbf{C}/\lambda)^{\top}u^{(t_{\text{in}})})$
    Update $\Delta v = \sum |v^{(t_{\text{in}})} - v^{(t_{\text{in}}-1)}|/N$
    **if** $\Delta v < \delta$ **then**
      Break
    **end if**
  **end for**
  Obtain optimal transport plan $\mathbf{T}_k^* = \text{diag}(u^{(t)}) \exp(-\mathbf{C}_k/\lambda)\text{diag}(v^{(t)})$
  Calculate the OT distance $d_{\text{OT}}(k) = \langle \mathbf{T}_k^*, \mathbf{C}_k \rangle$
  Calculate the classification probability $p_{\text{OT}}(y = k|x)$ with the OT distance
  Update the parameters of prompts $\{\omega_n\}_{n=1}^{N}$ with cross-entropy loss $L_{\text{CE}}$
**end for**
**return** Optimized prompt parameters $\{\omega_n\}_{n=1}^{N}$

## 6.3 Results

We evaluate our method against three standard benchmark datasets for Temporal Activity Localisation and report our results. Unless otherwise stated, we randomly select five samples for each class in each dataset, train for 100 epochs, and evaluate over the whole test set. In the few-shot setup, this is referred to as 5-shot, $C$-way. In THUMOS-14[102] this is 5-shot 20-way, for ActivityNet [28] it is 5-shot 200-way, and for EPIC-Kitchens[44] it is 5-shot 300-way for nouns and 5-shot-97-way for the verb partition.

### 6.3.1 Datasets

We have already introduced the THUMOS-14 and EPIC-Kitchens-100 datasets in the previous chapter, but in this chapter, we also evaluate the method on the ActivityNet dataset. ActivityNet 1.3 [28] is a large-scale dataset for human activity recognition and action localisation. It features approximately 20,000 untrimmed videos richly annotated across 200 action classes. This dataset is particularly noted for its realistic and diverse settings, capturing various human actions from daily life to sports and recreational activities. ActivityNet 1.3 provides over 23,000 temporal annotations, making it one of the most extensive datasets for TAL. The videos span a total of around 700 hours, with each video containing multiple actions. We test on ActivityNet to demonstrate how our method is effective in a range of environments and actions.

### 6.3.2 Comparative Methods

No current methods approach the few-shot temporal action localisation task with multiple prompt learning for each class. In [146], the authors present a multimodal setup with single prompt learning. However, they train the network in a meta-learning setup, train and test on disjoint sets, and use score fusion to correct misclassified segments (combining scores from UntrimmedNet [212]). Therefore, we assess our method's effectiveness against SOTA prompt learning frameworks and apply them to the task of TAL. First, for CoOP [264]- we initialize 16 learnable ctx tokens for each prompt. We also include two further baselines. Baseline $I$ removes the optimal transport component labelled section (e) in Figr̃effig:method by taking the mean of the $N$ learnable prompts to form one prompt for further processing in section (f). We also apply a linear probe (Baseline $II$ - LP) as outlined in [164] and [35] replacing sections (e) and (f) with local self-attention and a CNN layer directly to the pre-trained I3D embeddings.

### 6.3.3 Evaluation

This section evaluates our approach against existing methods for both few-shot temporal action localisation and prompt learning.

To compare with previous works, we report the mean average precision (mAP) at various intersections over union for all results.

**Table 6.1:** Performance comparison of our proposed method PLOT-TAL on the THUMOS-14 dataset against baselines.

| Method | mAP@0.3 | mAP@0.4 | mAP@0.5 | mAP@0.6 | mAP@0.7 | Avg (mAP) |
|---|---|---|---|---|---|---|
| Baseline I (avg) | 37.3 | 32.93 | 26.88 | 18.17 | 8.83 | 24.82 |
| Baseline II (lp) | 51.98 | 46.5 | 36.79 | 25.62 | 14.66 | 35.11 |
| CoOP | 48.73 | 43.67 | 36.64 | 27.24 | 16.97 | 34.65 |
| PLOT-TAL CLS | 53.46 | 48.93 | 38.2 | 30.2 | 18.8 | 38.24 |
| PLOT-TAL Verbose | **56.42** | **50.54** | **42.48** | **32.35** | **21.17** | **40.59** |

**THUMOS-14**

In Tab 6.1, we show results for 5-shot 20-way TAL on the THUMOS'14 dataset for our approach *PLOT-TAL CLS*. Adding additional class prompts can improve performance over a single prompt by a large margin ($\uparrow$ 5.9). We also show how it's possible to achieve higher accuracy by handcrafting prompts (Verbose). In this setting, we use GPT-3.5 [25] to produce additional descriptions of the actions that will replace the class label. Examples of the additional prompts are provided in Tab 6.10.

The Baseline $I$ method represents performance when we add additional prompts but exclude optimal transport, demonstrating how optimal transport is highly effective at aligning the features ($\uparrow$ 15.77). While Baseline $II$ based on the work of [164] and [35] has an average performance of 5% less than our method, demonstrating the importance of the sections (e) and (f).

In Fig 6.3, we demonstrate how the optimal transport improves performance at higher IoU thresholds than single prompt or linear probe methods.

At low IoU thresholds, the predicted segment only needs to overlap with a small section of the ground truth, meaning that single prompt methods and linear probes achieve relatively good performance as they distribute the attention between prompts and features across the temporal domain. However, as we increase the IoU threshold, we can see that our PLOT-TAL method becomes more effective, demonstrating the network's higher discriminative ability.

**EPIC-KITCHENS-100**

In Tab 6.2, we show results on the EPIC Kitchens verb and noun partitions, showing a slight improvement over single prompt methods for the noun classes ($\uparrow$ 1.19) but achieve a more significant performance boost for the verb classes ($\uparrow$ 2.96).

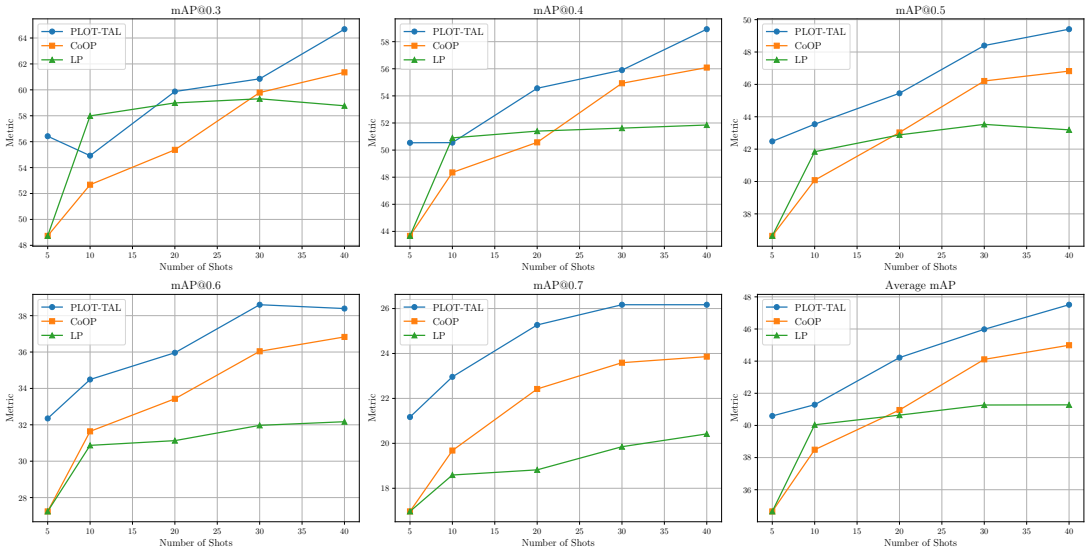



**Figure 6.3:** mAP over various IoU thresholds for the THUMOS-14 dataset. We show that the additional prompts improve performance over a single learnable prompt, as in CoOP.

| | Epic-Kitchens Noun (mAP) | | | | | | Epic-Kitchens Verb (mAP) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | Avg | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | Avg |
| Baseline I | 14.3 | 13.5 | 13.1 | 10.3 | 9.3 | 12.1 | 21.2 | 19.9 | 18.0 | 15.2 | 11.9 | 17.3 |
| Baseline II | **18.0** | 15.4 | 14.1 | 12.2 | 9.5 | 13.9 | **22.5** | **21.3** | 19.2 | 17.1 | 13.3 | 18.7 |
| CoOp | 16.1 | 15.0 | 13.8 | 11.8 | 9.5 | 13.3 | 18.5 | 17.6 | 16.3 | 14.6 | 12.5 | 15.9 |
| PLOT-TAL | 17.9 | **16.7** | **15.1** | **12.7** | **10.0** | **14.5** | 21.8 | 20.9 | **19.4** | **17.6** | **14.6** | **18.9** |

**Table 6.2:** Performance comparison on EPIC-Kitchens dataset for noun and verb recognition.

| Method | Approach | Shot/Way | Avg (mAP) |
|---|---|---|---|
| Common Action Localization [242] | ML | 5/5 | 22.8 |
| MUPPET [146] | ML + PL | 5/5 | 24.9 |
| Multi-Level Alignment [112] | ML | 5/5 | 31.8 |
| Query Adaptive Transformer [150] | ML | 5/5 | 32.7 |
| CoOP [265] | E2E + PL | 5/20 | 34.65 |
| PLOT TAL CLS | E2E + PL | 5/20 | **38.24** |
| PLOT-TAL Verbose | E2E + PL | 5/20 | **40.59** |

**Table 6.3:** Additional comparisons with existing Meta-Learning (ML), Prompt Learning (PL), and End to End (E2E) methods for few-shot temporal action localisation on the THUMOS'14 dataset.

| Method | mAP@0.5 |
|---|---|
| Hu et al. [93] | 45.4 |
| Yang et al. [242] | 56.5 |
| Yang et al. [243] | 60.6 |
| Nag et al. [150] | 63.0 |
| **PLOT-TAL CLS** | **65.1** |
| **PLOT-TAL Verbose** | **66.3** |

**Table 6.4:** Comparison with state-of-the-art methods for FS-TAL on ActivityNet1.3.

This demonstrates the effectiveness of the additional prompts in distinguishing between complex temporal features. However, the performance improvement is less pronounced for the noun partition. This suggests that nouns, which are generally static and visually distinct, are inherently easier to classify with a single prompt. As a result, they do not derive as much benefit from the added context provided by multiple prompts. Nouns typically represent objects with consistent visual appearances, reducing the need for additional context to disambiguate them. Therefore, the application of optimal transport, which excels in aligning distributions of more dynamic and context-dependent features (such as verbs), does not yield a substantial advantage in this case.

In Tab 6.3, we compare with other SOTA methods for few-shot temporal action localisation, which utilise meta-learning and perform few-shot localisation at a $5 - shot$, $5 - way$ setting, whereas our results are from the 5-shot, 20-way configuration. Not only is the 5-shot, 20-way few-shot setting more challenging, but PLOT-TAL also benefits from being trained end-to-end without the requirement for pre-training and episodic adaptive contrastive learning as in current meta-learning approaches.

### 6.3.4 Qualitative Results

In Fig 6.4, we show the normalised transport cost for each frame and $N$ embedding for the class label 'Cricket Shot'. This figure shows how each $N$ prompts diverge and focuses on different elements and views within the videos. For example, we can see that $N_1$ or Prompt 1 learns global information across all frames. This shows how we may distribute alignment across all frames in a single prompt framework and lose discriminative ability since it learns global information over the whole video. In the figure, we can note that Prompt 4 appears to learn background information and is more closely aligned to frames where we can see the stadium stands. Prompts


**Figure 6.4:** The normalised transport cost of each $N$ prompt for the class 'Cricket Shot' after training. Prompt one aligns with global information, while the other prompts learn additional, complementary views. In the transport cost algorithm, a lower value indicates closer alignment.

2 and 3, however, indicate a closer alignment with objects related to the class of 'cricket shot,' including when the cricket strip is in the shot and there are people on the field. The transport costs in the plot are normalized according to the global maximum and minimum of all transport scores across the different prompts. While this normalization means that the absolute values are not directly comparable between prompts, it serves to place all prompts on a common scale for visualization purposes. Despite this, the plot clearly reveals that each prompt, when considered in isolation, demonstrates varying degrees of alignment with different sections of the video. This suggests that each prompt is capturing unique and complementary aspects of the video content, allowing for a more nuanced understanding of the temporal dynamics. The distinct patterns observed across prompts indicate that, even after normalization, the prompts retain their ability to differentiate between various segments of the video, highlighting the effectiveness of our approach.

### 6.3.5 Ablation Experiments

We perform several ablation experiments to evaluate each component of the architecture. We experiment with the number of learnable context tokens and prompts per class, alternative feature

| N prompts | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | avg |
|---|---|---|---|---|---|---|
| N=4 | 55.88 | 50.21 | **43.06** | 31.97 | 21.16 | 40.46 |
| N=6 | **56.42** | **50.54** | 42.48 | 32.35 | 21.17 | **40.59** |
| N=8 | 53.60 | 48.72 | 41.74 | 31.68 | 20.70 | 39.29 |
| N=10 | 54.96 | 50.27 | 43.45 | **32.53** | **21.44** | 40.53 |
| N=12 | 53.74 | 48.25 | 41.02 | 30.57 | 20.06 | 38.73 |
| N=14 | 54.25 | 48.94 | 40.90 | 30.78 | 18.86 | 38.75 |
| N=16 | 53.66 | 48.28 | 41.04 | 30.84 | 20.15 | 38.79 |

**Table 6.5:** Ablation experiment varying the number of additional learnable prompts for each class.

alignment metrics, and the number of feature pyramid network levels. We also experimented with the types of RGB embeddings and several prompt-engineering strategies.

**Number of Learnable Prompts**

In Tab 6.5, we perform an ablation experiment on the number of learnable prompts $N$. The results show that the optimum number of prompts is $N = 6$, while with an increased number of prompts, e.g., $N = 10$, we can achieve better results in the more difficult IoU thresholds. This is due to the increased temporal discriminative ability of the additional prompts. As the N increases, performance degrades as the model overfits due to the increased number of learnable parameters.

**Number of Learnable Context Tokens**

Each prompt also has several learnable context tokens as described in [260] and [223]. These context tokens are randomly initialised so that for the class 'Basketball Dunk' with 4 $ctx$ tokens, the full prompt will be

$$P = \{X, X, X, X, \text{Basketball Dunk}\} \tag{6.13}$$

In Tab 6.6, we show the effect of varying the number of learnable $ctx$ tokens appended to each prompt. For each $N$ prompt, $n_{ctx}$ tokens are randomly initialised. The figure shows that the optimum number of tokens is between 10 and 20. As per the existing literature [260, 264], we select 16 tokens for all methods unless otherwise stated and train and test using the 5-shot, 20-way setup.

**Table 6.6:** Ablation experiment on the number of context tokens on the THUMOS'14 Dataset.

| Ctx Tokens | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | avg |
|---|---|---|---|---|---|---|
| 1 | 52.25 | 46.94 | 40.73 | 31.26 | 20.17 | 38.27 |
| 10 | 54.94 | 49.55 | **42.49** | 31.14 | 20.08 | 39.64 |
| 16 | **56.42** | **50.54** | 42.48 | 32.35 | **21.17** | **40.59** |
| 20 | 53.39 | 48.38 | 42.19 | **33.00** | 20.78 | 39.55 |
| 30 | 50.27 | 45.54 | 38.30 | 29.64 | 18.83 | 36.52 |
| 40 | 53.55 | 47.30 | 40.35 | 31.06 | 19.46 | 38.34 |

**FPN Levels**

In Tab 6.7, we show the effect of increasing or decreasing the number of feature pyramid levels in the network. The results show that six is the optimum number. Additional FPN layers beyond six will tend to increase the number of parameters for optimisation while not providing any additional benefit.

**Feature Matching Strategy**

To assess the efficacy of using Optimal Transport (OT) with the Sinkhorn Algorithm to align video features with adaptive prompts, we conducted ablation experiments in which OT was replaced with more straightforward distance metrics, precisely Euclidean distance and Hungarian distance. Our goal was to determine the impact of these substitutions on alignment performance and overall method effectiveness.

**Euclidean Distance**

We replaced the OT metric with the Euclidean distance in the first variant. Here, the alignment between the refined video features $\{\mathbf{x'_1}, \mathbf{x'_2}, \ldots, \mathbf{x'_t}\}$ and the adaptive prompts $\mathbb{P}_k$ for each action category $k$ was performed directly using the Euclidean distance:

$$d_{\text{Euc}}(\mathbf{u}, \mathbf{v_k}) = \sum_{t=1}^{T} \sum_{i=1}^{N} \|\mathbf{x'_t} - \mathbf{p_{ki}}\|^2$$

In this formulation, the cost matrix $C_{ti}$ is defined as the squared Euclidean distance between video feature $\mathbf{x'_t}$ and prompt embedding $\mathbf{p_{ki}}$:

$$C_{ti} = \|\mathbf{x'_t} - \mathbf{p_{ki}}\|^2$$

The alignment process involves directly computing the sum of these distances without optimising a transport plan.

**Hungarian Distance**

In the second variant, we utilised the Hungarian algorithm to find an optimal one-to-one matching between video features and prompts, minimising the overall distance. The cost matrix $C_{ti}$ is defined similarly to the Euclidean distance case, but the Hungarian algorithm ensures a unique assignment of each video feature to a prompt:

$$d_{\text{Hung}}(\mathbf{u}, \mathbf{v_k}) = \min_{\mathbf{T} \in \Pi} \sum_{t=1}^{T} \sum_{i=1}^{N} C_{ti} T_{ti} \tag{6.14}$$

Here, $\Pi$ represents the set of all possible permutations that allow a one-to-one matching between the sets of video features and prompts.

In Tab 6.8, we show that OT outperforms both methods. The superior performance of OT can be attributed to several key factors:

- **Global Distribution Matching**: OT aligns the entire distribution of video features with the prompts distribution, considering the global structure and interdependencies within the data. In contrast, Euclidean distance considers each pair independently, which can lead to suboptimal alignments in the presence of complex feature distributions.
- **Flexible Many-to-Many Matching**: OT allows for a many-to-many correspondence between video features and prompts, providing more flexibility in the alignment process. On the other hand, the Hungarian algorithm enforces a strict one-to-one matching, which may not capture the underlying relationships effectively, significantly when the number of video features and prompts differ significantly.
- **Entropic Regularization**: The Sinkhorn algorithm introduces entropic regularisation, promoting smoother and more stable solutions by avoiding challenging assignments. This regularisation helps mitigate the impact of noisy or outlier features, leading to more robust alignments.

**Visual Feature Embeddings**

| FPN Network Levels | 0.5 | avg (mAP) |
|---|---|---|
| 1 | 25.82 | 26.16 |
| 2 | 37.80 | 35.81 |
| 3 | 39.10 | 36.58 |
| 4 | 40.02 | 38.03 |
| **5** | **43.06** | **40.46** |
| 6 | 42.21 | 39.57 |
| 7 | 41.56 | 38.92 |

**Table 6.7:** Ablation experiment varying the number of FPN levels with 0.5 and average (mAP) values.

| Method | 0.5 | avg (mAP) |
|---|---|---|
| Euclidean | 21.97 | 22.27 |
| Kuhn-Munkres | 29.48 | 29.09 |
| OT | 43.06 | 40.46 |

**Table 6.8:** Experiment comparing various prompt alignment methods.

To evaluate the effectiveness of adding motion information via optical flow, we also performed additional experiments using only the RGB embeddings, the optical flow embeddings, and RGB CLIP embeddings from a ViT-B-16 encoder, with results shown in Tab 6.9. The results show that the CLIP embeddings perform better than the RGB from the I3D network $\uparrow 2.67$. This is because of the implicit alignment between the image and text encoder embeddings before temporal convolution. However, when combined with optical flow, the performance is improved by a large margin of $\uparrow 7.56$, demonstrating the enhanced classification ability of the network when we add additional temporal information via optical flow.

**Table 6.9:** Comparison of mAP scores for various visual input embeddings on the THUMOS'14 dataset.

| **Embeddings** | **0.3** | **0.4** | **0.5** | **0.6** | **0.7** | **avg (mAP)** |
|---|---|---|---|---|---|---|
| CLIP | 46.99 | 42.09 | 34.26 | 25.34 | 15.82 | 32.90 |
| RGB | 43.13 | 38.76 | 31.71 | 23.15 | 14.46 | 30.24 |
| Optical Flow | 26.03 | 23.10 | 19.54 | 14.07 | 8.93 | 18.33 |
| RGB + Flow | **55.88** | **50.21** | **43.06** | **31.97** | **21.16** | **40.46** |

**Prompt Engineering**

**Table 6.10:** GPT generated descriptions for PLOT-TAL Verbose on THUMOS'14 Dataset.

| ID | Description |
|---|---|
| 7 | The precise moment a baseball player winds up and releases the ball towards the batter |
| 9 | The instant a basketball player leaps into the air to forcefully slam the ball through the hoop |
| 12 | The exact moment the cue stick strikes the cue ball, initiating the billiards shot |
| 21 | The moment a weightlifter hoists the barbell from the ground to overhead in one fluid motion |
| 22 | The split second a diver leaps off the cliff edge, beginning their descent into the water below |
| 23 | The moment a cricket bowler releases the ball towards the batsman with a swift arm motion |
| 24 | The precise moment the batsman swings the bat to strike the cricket ball |
| 26 | The instant a diver jumps off the board, tucking and twisting before plunging into the pool |
| 31 | The moment a frisbee is caught by a leaping player, securing it firmly in their hands |
| 33 | The exact moment a golfer swings the club, making contact with the ball to send it flying |
| 36 | The moment an athlete spins and releases the hammer, propelling it into the air |
| 40 | The split second an athlete takes off over the high jump bar, attempting to clear it without touching |
| 45 | The precise moment the javelin is thrown, with the athlete's arm extending forward in a powerful motion |
| 51 | The instant an athlete sprints and leaps into the air to cover the maximum distance before landing in the sand pit |
| 68 | The moment an athlete plants the pole in the box and vaults over the bar, pushing themselves upwards |
| 79 | The exact moment the shot is put from the neck, using one hand, in a pushing motion through the air |
| 85 | The moment a soccer player strikes the ball with their foot aiming to score a penalty kick |
| 92 | The precise moment a tennis player swings their racket to strike the incoming ball |
| 93 | The instant an athlete spins and releases the discus, hurling it into the designated sector |
| 97 | The moment a volleyball player jumps and forcefully spikes the ball over the net towards the opponent's court |

We also demonstrate how crafted prompts can help boost performance as per a prompt-engineering setup. In Tab 6.10, we show the prompts generated by GPT 3.5 with the prompt - *'Generate prompts for a temporal action localisation task for the following class IDs. The prompts should include objects, the action, and some indication of the moment when the action occurs. We anticipate that further prompt engineering strategies will yield improved results.*

## 6.4 Conclusion

This chapter introduced an efficient approach for training a vision-language network to perform action localisation with minimal annotations. We demonstrated that aligning multiple prompts across various temporal resolutions enhances the network's few-shot learning capabilities. Crucially, reducing the number of learnable parameters improves generalisation and optimises training efficiency. This method, requiring only five annotated examples per class, is well-suited for applications in video understanding where data labelling resources are scarce or in specialised contexts such as sports analysis, where precise action localisation is critical.

However, this methodology is not without limitations. It still necessitates a minimal amount of labelled data, preventing its application in zero-shot scenarios where no labelled data is available. Furthermore, the Optimal Transport Map integration incurs a slight increase in training and inference times—approximately 0.02 seconds per sample. Looking ahead, we anticipate significant advancements in contrastive language video models, particularly in training on a broader and more diverse range of content. Such improvements are expected to enhance zero-shot localisation capabilities by providing richer representations. The proposed method could readily incorporate these enhancements, potentially achieving more accurate initial alignments with text embeddings, yielding substantial performance gains.

This chapter concludes the main body of the thesis. The subsequent chapter will summarise the contributions presented throughout the thesis and explore potential avenues for future research.

# Chapter 7

# Conclusions and Future Work

Throughout this thesis, we explored several deep-learning strategies for multimodal video understanding, particularly in environments with limited data and computational resources. Our research has demonstrated the utility of leveraging large pre-trained networks to extract multimodal features. However, we have also identified the need for application-specific approaches to fuse these features effectively. The techniques developed have proven to enhance recommendation systems, clustering, classification, and action localisation tasks without requiring substantial GPU resources or extensive datasets. Notably, all methods were optimised to run on a single consumer-grade GPU, highlighting the practicality and accessibility of our approaches for organisations who wish to implement advanced video understanding tools at minimal cost.

This work contributes to the broader field of video understanding by providing efficient solutions that reduce the barriers to entry for advanced video analysis technologies. These contributions advance the technical capabilities of handling video data and democratise video understanding tools, making them accessible to a broader range of users and applications.

## 7.1 Conclusions

Overall, the objectives of this thesis were to:

- Explore new methodologies and approaches for efficient video understanding tasks using deep learning.

- Identify methods for fusing multimodal and spatio-temporal features that improve over uni-modal methods.
- Introduce techniques for long-video understanding applications which are computationally and data-efficient, reducing the requirement for extensive annotation.

We achieved these goals throughout the following chapters.

In Chapter 3, we introduced a method for fusing multiple modalities from video for style and semantic clustering of videos with limited labels. Extracting features from several expert foundational models enabled the model to learn contextual features that offered more stylistic clues than metadata alone. We also introduced a new dataset that covered an extensive range of global cinema history to evaluate our method. The method presented could be used to build new video recommendation systems that could be implemented in video archives without data labelling. Such a method has broader applications in media studies where new semantic and stylistic elements could be compared between periods of cinematic history.

Chapter 4 tackled the problem of long video understanding with limited resources. Again, we demonstrated how to leverage pre-trained encoders to extract spatio-temporal features at different resolutions. We were able to train a network efficiently by leveraging the inductive bias of the spatial stream while adding temporal understanding through the lightweight temporal encoder. This architecture demonstrated SOTA results on tasks requiring more fine-grained temporal understanding, including speaker recognition and character identification.

In Chapter 5, we introduce the problem of audio-visual fusion for temporal action localisation and introduce gated cross-attention with visual context, ensuring that only useful audio information is included in the regression and classification tasks. The network uses pre-extracted features and a simple pooling mechanism in the feature pyramid to ensure a low-parameter and efficient solution to this problem.

Finally, in Chapter 6, we presented a novel method for aligning text prompts with pre-extracted visual features using prompt learning and optimal transport. In this example, we demonstrated how optimal transport could efficiently align features while the additional learned prompts could be used to localise and discriminate between foreground and background views within the video using only a few labelled samples per class.

## 7.2 Future Research Directions

Several limitations exist outside the scope of this thesis that are nonetheless important for the future of accessible video understanding deep learning methods.

**Model Efficiency**

The first is that the architectures presented cannot process video in real-time, a common requirement in many applications such as sports analysis, healthcare, and robotics. Recent advancements include adaptive network architectures that can adapt to hardware requirements during training [66], multimodal memory caching [222], alongside extensions of quantisation and pruning [190, 204]. Recently, novel state-space models such as MAMBA have shown favourable efficiencies directly relevant for long-form video understanding [120]. A future approach could focus on developing federated learning techniques for video data. This would allow models to be trained across multiple decentralised devices, reducing the need for high computational power and bandwidth while maintaining privacy and improving real-time analysis capabilities. Such a distributed approach can be particularly transformative for applications in healthcare and urban monitoring, where data sensitivity is paramount.

**Advancements in Zero-Shot Learning for Video**

As discussed in Chapter 6, zero-shot learning has the potential to advance the accessibility of video understanding systems where labelled data is scarce or expensive to obtain. Chapter 6 demonstrated a method for adapting CLIP with prompt learning and alignment for few-shot temporal action localisation. Recent works have begun demonstrating advances in the zero-shot setup. These include using more expressive and powerful language models [138], improvements in existing video-language contrastive pre-training [217], and access to more extensive and well-annotated datasets [263] incorporating more diverse views and modalities [77]. An exciting future direction is exploring synthetic data generation for zero-shot learning. Using advanced generative models to create realistic and diverse video scenarios, researchers can train video understanding systems that can generalise better without requiring extensive labelled datasets. This could include the development of synthetic actors and environments that provide a richer array of training examples for zero-shot video understanding models.

**Cross-Modal Coherence and Synchronisation**

Achieving effective synchronisation between modalities such as audio, video, and textual descriptions in a unified framework remains challenging. Current research continues to explore scaling foundation models with multimodal data [217] while more efficient strategies include improving feature alignment during contrastive pre-training [251]. An emergent area of research is also video quality assessment and multimodal alignment for training data. This is essential for current training paradigms of multimodal generative video networks which require aligned and unedited video. A future research area could involve the application of blockchain technology to validate and synchronise multimodalities in a secure manner. This could ensure the integrity and alignment of data used in training multimodal systems, particularly in scenarios where data comes from disparate and potentially untrustworthy sources. Further, exploring the integration of causal inference methods to better understand and model the interactions between different modalities could lead to more robust synchronisation within video understanding systems.

# Bibliography


[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A Large-Scale Video Classification Benchmark. 2016.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[3] Juan León Alcázar, Fabian Caba, Ali K Thabet, and Bernard Ghanem. Maas: Multi-modal assignation for active speaker detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 265–274, 2021.

[4] Rick Altman. 3 . A Semantic / Syntactic Approach to film genre. *Cinema Journal*, 23(3):6–18, 1984.

[5] Federico Álvarez, Faustino Sánchez, Gustavo Hernández-Peñaloza, David Jiménez, José Manuel Menéndez, and Guillermo Cisneros. On the influence of low-level visual features in film classification. *PloS one*, 14(2):e0211406, 2019.

[6] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3173–3183, 2021.

[7] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2018.

[8] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV-17*, pages 609–617, 2017.

[9] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[10] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *International Conference on Computer Vision*, 2021.

[11] Aida Austin, Elliot Moore, Udit Gupta, and Parag Chordia. Characterization of movie genre based on music score. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 421–424. IEEE, 2010.

[12] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*. Springer, 2011.

[13] Anurag Bagchi, Jazib Mahmood, Dolton Fernandes, and Ravi Kiran Sarvadevabhatla. Hear me out: Fusional approaches for audio augmented temporal action localization. *arXiv preprint arXiv:2106.14118*, 2021.

[14] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 121–137. Springer, 2020.

[15] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[16] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019.

[17] Olfa Ben-Ahmed and Huet Benoit. Deep multimodal features for movie genre and interestingness prediction. In *International Conference on Content-Based Multimedia Indexing (CBMI)m*, 2018.

[18] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases (KDD)*, pages 359–370, 1994.

[19] Erik Bernhardsson. Annoy: Approximate nearest neighbors in c++/python, 2018. Version 1.17.0.

[20] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *Proceedings of the International Conference on Machine Learning*, pages 813–824, 2021.

[21] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[22] Aashish Bhandari, Siddhant B Shah, Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. Crisishatemm: Multimodal analysis of directed and undirected hate speech in text-embedded images from russia-ukraine conflict. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1993–2002, 2023.

[23] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[24] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.

[25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[26] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017.

[27] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[28] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[29] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[30] Joao Carreira and Andrew Zisserman. 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, 2017.

[31] Joao Carreira and Andrew Zisserman. 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, 2017.

[32] Paola Cascante-Bonilla, Kalpathy Sitaraman, Mengjia Luo, and Vicente Ordonez. Moviescope: Large-scale analysis of movies using multiple modalities. *arXiv preprint arXiv:1908.03180*, 2019.

[33] Shuning Chang, Pichao Wang, Fan Wang, Hao Li, and Jiashi Feng. Augmented transformer with adaptive graph for temporal action proposal generation. *arXiv preprint arXiv:2103.16024*, 2021.

[34] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[35] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022.

[36] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. Dcan: improving temporal action detection via dual context aggregation. In *AAAI*, 2022.

[37] Jingzhou Chen and Shin'ichi Satoh. Deep cross-modal audio-visual generation. In *Thirteenth Annual ACM International Conference on Multimedia*, 2017.

[38] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[39] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:89–113, 2021.

[40] Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with long-memory transformer. *European Conference on Computer Vision*, 2022.

[41] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26, 2013.

[42] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693. PMLR, 2014.

[43] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. *European conference on computer vision*, 2006.

[44] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[45] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[46] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2021.

[47] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[48] Chaorui Deng, Qi Chen, Pengda Qin, Da Chen, and Qi Wu. Prompt switch: Efficient clip adaptation for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15648–15658, 2023.

[49] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR-09*, 2009.

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[52] Piotr Dollar et al. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance*, 2005.

[53] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[54] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convo-

lutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[55] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Word2visualvec: Image and video to sentence matching by visual feature prediction. *arXiv preprint arXiv:1604.06838*, 2016.

[56] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[57] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. *Proceedings of the European Conference on Computer Vision*, pages 751–767, 2000.

[58] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018.

[59] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 768–784. Springer, 2016.

[60] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. *Proceedings of the Scandinavian Conference on Image Analysis*, pages 363–370, 2003.

[61] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[62] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[63] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[64] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR-17*, pages 3636–3645, 2017.

[65] Edward Fish, Jon Weinbren, and Andrew Gilbert. Rethinking genre classification with fine grained semantic clustering. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021.

[66] Lin Geng Foo, Jia Gong, Zhipeng Fan, and Jun Liu. System-status-aware adaptive network for online streaming video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10514–10523, 2023.

[67] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[68] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 214–229. Springer, 2020.

[69] Alfred Galichon. *Optimal transport methods in economics*. Princeton University Press, 2018.

[70] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[71] Jacob Gildenblat and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

[72] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *CoRR*.

[73] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *arXiv preprint arXiv:2106.02036*, 2021.

[74] Guoqiang Gong, Liangfeng Zheng, and Yadong Mu. Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[75] Vaishali U Gongane, Mousami V Munot, and Alwin D Anuse. Detection and moderation of detrimental content on social media platforms: Current status and future directions. *Social Network Analysis and Mining*, 12(1):129, 2022.

[76] Robert Gorwa, Reuben Binns, and Christian Katzenbach. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1):2053951719897945, 2020.

[77] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. *arXiv preprint arXiv:2311.18259*, 2023.

[78] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. *arXiv preprint arXiv:2107.06263*, 2021.

[79] Alexander G Hauptmann, Rong Jin, and Tobun Dorbin Ng. Multi-modal information retrieval from broadcast video using ocr and speech recognition. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 160–161, 2002.

[80] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR-16*, pages 770–778, 2016.

[81] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[82] Yilong He, Yong Zhong, Lishun Wang, and Jiachen Dang. Glformer: Global and local context aggregation network for temporal action detection. *Applied Sciences*, 12(17):8557, 2022.

[83] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1914–1923, 2016.

[84] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee International Conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[85] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[86] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv/1912.12180*, 2019.

[87] Chiori Hori, Takaaki Hori, Teng-Yok Lee, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4203–4212, 2017.

[88] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4193–4202, 2017.

[89] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[90] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[91] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR-18*, pages 7132–7141, 2018.

[92] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[93] Tao Hu, Pascal Mettes, Jia-Hong Huang, and Cees GM Snoek. Silco: Show a few images, localize the common object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5067–5076, 2019.

[94] Hui-Yu Huang, Weir-Sheng Shih, and Wen-Hsing Hsu. A film classifier based on low-level visual features. In *2007 IEEE 9th Workshop on Multimedia Signal Processing*, pages 465–468. IEEE, 2007.

[95] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European Conference on Computer Vision*. Springer, 2020.

[96] Yin-Fu Huang and Shih-Hao Wang. Movie genre classification using svm with audio and video features. In *International Conference on Active Media Technology*, pages 1–10. Springer, 2012.

[97] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[98] Sanjay K Jain and RS Jadon. Movies genres classifier using neural network. In *2009 24th International Symposium on Computer and Information Sciences*, pages 575–580. IEEE, 2009.

[99] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society Conference on Computer Vision and Pattern Recognition*, 2010.

[100] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[101] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014.

[102] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes, 2014.

[103] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022.

[104] Simon J Julier and Jeffrey K Uhlmann. A new extension of the kalman filter to nonlinear systems. *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, 3:182–193, 1997.

[105] Guoliang Kang, Liang Liang, Kris M Kitani, and Takeo Kanade. A deep learning model with late fusion for person re-identification based on intra-class distance. In *International Conference on Computer Vision*, 2016.

[106] Tae-Kyung Kang, Gun-Hee Lee, and Seong-Whan Lee. Htnet: Anchor-free temporal action localization with hierarchical transformers. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 365–370. IEEE, 2022.

[107] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[108] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[109] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[110] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. *arXiv preprint arXiv:2111.01024*, 2021.

[111] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019.

[112] Kanchan Keisham, Amin Jalali, Jonghong Kim, and Minho Lee. Multi-level alignment for few-shot temporal action localization. *Information Sciences*, 650:119618, 2023.

[113] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[114] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[115] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[116] Alexander Kläser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, 2008.

[117] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Neural Information Processing Systems-18*, pages 7763–7774, 2018.

[118] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

[119] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963, 2022.

[120] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024.

[121] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11499–11506, 2020.

[122] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2021.

[123] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[124] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019.

[125] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[126] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[127] Hong Liu, Guanghui Wang, and Zhenhua Hu. Cross-modal video retrieval: A benchmark and baseline. *Information Sciences*, 460:292–304, 2018.

[128] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11612–11619, 2020.

[129] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[130] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip HS Torr. Multi-shot temporal event localization: a benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12596–12606, 2021.

[131] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31:5427–5441, 2022.

[132] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *BMVC-19*, 2019.

[133] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[134] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019.

[135] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[136] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 81, pages 674–679, 1981.

[137] Gregory Luklow and Steven Ricci. The ”audience” goes ”public”: Inter-textuality, genre, and the responsibilities of film literacy. (12):29, 1984.

[138] Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. Zero-shot video moment retrieval from frozen vision-language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5464–5473, 2024.

[139] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[140] Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from video and text via large-scale discriminative clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5267–5276, 2017.

[141] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017.

[142] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017.

[143] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with Context Gating for video classification, jun 2017.

[144] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.

[145] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 19–27, 2018.

[146] Sauradip Nag, Mengmeng Xu, Xiatian Zhu, Juan-Manuel Pérez-Rúa, Bernard Ghanem, Yi-Zhe Song, and Tao Xiang. Multi-modal few-shot temporal action detection via vision-language meta-adaptation. *arXiv preprint arXiv:2211.14905*, 2022.

[147] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Post-processing temporal action detection. *arXiv preprint arXiv:2211.14924*, 2022.

[148] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Proposal-free temporal action detection via global segmentation mask learning. In *Computer Vision–ECCV 2022: 17th*

*European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 645–662. Springer, 2022.

[149] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *European Conference on Computer Vision*, pages 681–697. Springer, 2022.

[150] Sauradip Nag, Xiatian Zhu, and Tao Xiang. Few-shot temporal action localization with query adaptive transformer. *arXiv preprint arXiv:2110.10552*, 2021.

[151] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *arXiv preprint arXiv:2107.00135*, 2021.

[152] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*, 2021.

[153] Steve Neale. Questions of Genre. *Film Genre Reader IV*, (July):178 – 202, 2012.

[154] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

[155] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV-18*, pages 631–648, 2018.

[156] Samuel Palazzo, Chris Peters, and Abdenour Hadid. Hybrid fusion for video summarization: A comprehensive review. *Journal of Image and Vision Computing*, 65:45–58, 2017.

[157] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[158] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[159] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017.

[160] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 485–494, 2021.

[161] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, 2017.

[162] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.

[163] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[164] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[165] Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE signal processing magazine*, 34(6):96–108, 2017.

[166] Merey Ramazanova, Victor Escorcia, Fabian Caba, Chen Zhao, and Bernard Ghanem. Owl (observe, watch, listen): Audiovisual temporal context for localizing actions in egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4879–4889, 2023.

[167] Zeeshan Rasheed, Yaser Sheikh, and Mubarak Shah. On the use of computable features for film classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):52–64, 2005.

[168] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pages 1–23, 2018.

[169] B. Reddy and A. Jadhav. Comparison of scene change detection algorithms for videos. In *2015 Fifth International Conference on Advanced Computing Communication Technologies*, pages 84–89, 2015.

[170] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[171] Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, 1995.

[172] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[173] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[174] Sreemanananth Sadanand and Jason J Corso. Action bank: A high-level representation of activity in video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[175] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Neural Information Processing Systems-17*, 2017.

[176] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM International Conference on Multimedia*, 2007.

[177] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via

gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[178] Prashant Giridhar Shambharkar, MN Doja, Dhruv Chandel, Kartik Bansal, and Kunal Taneja. Multimodal kdk classifier for automatic classification of movie trailers. *IJRTE*, 2019.

[179] Prashant Giridhar Shambharkar, Pratyush Thakur, Shaikh Imadoddin, Shantanu Chauhan, and MN Doja. Genre classification of movie trailers using 3d convolutional neural networks. *ICICCS 2020*, 2020.

[180] Dingfeng Shi, Qiong Cao, Yujie Zhong, Shan An, Jian Cheng, Haogang Zhu, and Dacheng Tao. Temporal action localization with enhanced instant discriminability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[181] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative boundary modeling. *arXiv preprint arXiv:2303.07347*, 2023.

[182] Joana Silva, João Madureira, Cláudia Tonelo, Daniela Baltazar, Catarina Silva, Anabela Martins, Carlos Alcobia, and Inês Sousa. Comparing machine learning approaches for fall risk assessment. In *International Conference on Bio-inspired Systems and Signal Processing*, volume 5, pages 223–230. SciTePress, 2017.

[183] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[184] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems*, 27, 2014.

[185] E.T. Smith and S.M. Kosslyn. Enhanced auditory capacities in visually impaired individuals. *Journal of Neuroscience*, 27(19):1234–1240, 2007.

[186] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu. Class semantics-based attention for action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13739–13748, 2021.

[187] Chris Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 246–252. IEEE, 1999.

[188] Chen Sun, Alan Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *Proceedings of the IEEE International Conference on Computer Vision*, pages 7464–7473, 2019.

[189] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[190] Ximeng Sun, Rameswar Panda, Chun-Fu Richard Chen, Aude Oliva, Rogerio Feris, and Kate Saenko. Dynamic network quantization for efficient video inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7375–7385, 2021.

[191] Javier Sánchez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. *Image Processing On Line*, 3:137–150, 07 2013.

[192] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13526–13535, 2021.

[193] Tuan N Tang, Kwonyoung Kim, and Kwanghoon Sohn. Temporalmaxer: Maximize temporal context with only max pooling for temporal action localization. *arXiv preprint arXiv:2303.09055*, 2023.

[194] George Thomas, Christopher McMahan, and Ronald Metoyer. Deep learning-based fusion of spatial data and text in sports analytics. *Journal of Sports Analytics*, 3(1):65–75, 2017.

[195] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 247–263, 2018.

[196] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *International Conference on Computer Vision*, 2019.

[197] Luis Caicedo Torres, Luiz Manella Pereira, and M Hadi Amini. A survey on optimal transport for machine learning: Theory and applications. *arXiv preprint arXiv:2106.01963*, 2021.

[198] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*. PMLR, 2021.

[199] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[200] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[201] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[202] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[203] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR-18*, pages 6450–6459, 2018.

[204] Ties van Rozendaal, Tushar Singhal, Hoang Le, Guillaume Sautiere, Amir Said, Krishna Buska, Anjuman Raha, Dimitris Kalatzis, Hitarth Mehta, Frank Mayer, et al. Mobilenvc: Real-time 1080p neural video compression on a mobile device. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4323–4333, 2024.

[205] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[206] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[207] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV-15*, pages 4534–4542, 2015.

[208] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[209] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.

[210] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[211] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[212] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[213] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[214] Lining Wang, Haosen Yang, Wenhao Wu, Hongxun Yao, and Hujie Huang. Temporal action proposal generation with transformers. *arXiv preprint arXiv:2105.12043*, 2021.

[215] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705, 2020.

[216] Xiaofang Wang, Xuehan Xiong, Maxim Neumann, AJ Piergiovanni, Michael S Ryoo, Anelia Angelova, Kris M Kitani, and Wei Hua. Attentionnas: Spatiotemporal attention cell search for video classification. In *European Conference on Computer Vision*, pages 449–465. Springer, 2020.

[217] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024.

[218] Jônatas Wehrmann and Rodrigo C Barros. Movie genre classification: A multi-label approach based on convolutions through time. *Applied Soft Computing*, 61:973–982, 2017.

[219] Jonatas Wehrmann, Rodrigo C Barros, Gabriel S Simoes, Thomas S Paula, and Duncan D Ruiz. (deep) learning from frames. In *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*, pages 1–6. IEEE, 2016.

[220] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR-18*, pages 8052–8060, 2018.

[221] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Department of Computer Science, 1995.

[222] Wujun Wen, Yunheng Li, Zhuben Dong, Lin Feng, Wanxiao Yang, and Shenlan Liu. Streaming video temporal action segmentation in real time. In *2023 18th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 316–323. IEEE, 2023.

[223] Yuetian Weng, Zizheng Pan, Mingfei Han, Xiaojun Chang, and Bohan Zhuang. An efficient spatio-temporal pyramid transformer for action detection. In *European Conference on Computer Vision*, 2022.

[224] Gert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *European conference on computer vision*, 2008.

[225] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021.

[226] Shikui Wu, Hau San Wong, and Zhiwen Yu. Multi-stream deep networks for person to person violence detection in videos. *Pattern Recognition*, 57:233–247, 2016.

[227] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 461–470, 2015.

[228] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[229] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018.

[230] Huijuan Xu, Ximeng Sun, Eric Tzeng, Abir Das, Kate Saenko, and Trevor Darrell. Revisiting few-shot activity detection with class similarity control. *arXiv preprint arXiv:2004.00137*, 2020.

[231] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR-16*, pages 5288–5296, 2016.

[232] Liang Xu and Zhiwen Wu. Learning multi-modal density maps for crowd counting. *IEEE Transactions on Multimedia*, 19(10):2328–2339, 2017.

[233] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020.

[234] Natsuo Yamamoto, Jun Ogata, and Yasuo Ariki. Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[235] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, volume 92, pages 379–385, 1992.

[236] Baosong Yang, Zhaopeng Tu, Derek F Wong, Fandong Meng, Lidia S Chao, and Tong Zhang. Modeling localness for self-attention networks. *arXiv preprint arXiv:1810.10182*, 2018.

[237] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020.

[238] Hongtao Yang, Xuming He, and Fatih Porikli. One-shot action localization by learning sequence matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1450–1459, 2018.

[239] Le Yang, Junwei Han, Tao Zhao, Nian Liu, and Dingwen Zhang. Structured attention composition for temporal action localization. *IEEE Transactions on Image Processing*, 2022.

[240] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29:8535–8548, 2020.

[241] Min Yang, Guo Chen, Yin-Dong Zheng, Tong Lu, and Limin Wang. Basictad: an astounding rgb-only baseline for temporal action detection. *arXiv preprint arXiv:2205.02717*, 2022.

[242] Pengwan Yang, Vincent Tao Hu, Pascal Mettes, and Cees GM Snoek. Localizing the common action among a few videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 505–521. Springer, 2020.

[243] Pengwan Yang, Pascal Mettes, and Cees GM Snoek. Few-shot transformation of common actions into time and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16031–16040, 2021.

[244] Guangnan Ye, Yitong Li, Hongtao Xu, Dong Liu, and Shih-Fu Chang. Eventnet: A large scale structured concept library for complex event detection in video. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 471–480, 2015.

[245] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[246] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[247] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007.

[248] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3643–3653, 2018.

[249] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019.

[250] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 492–510. Springer, 2022.

[251] Ming Zhang, Ke Chang, and Yunfang Wu. Multi-modal semantic understanding with contrastive cross-modal feature alignment. *arXiv preprint arXiv:2403.06355*, 2024.

[252] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[253] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021.

[254] Zhongping Zhang, Yiwen Gu, Bryan A Plummer, Xin Miao, Jiayi Liu, and Huayan Wang. Movie genre classification by language augmentation and shot sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7275–7285, 2024.

[255] Chen Zhao, Ali K Thabet, and Bernard Ghanem. Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13658–13667, 2021.

[256] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 539–555. Springer, 2020.

[257] Rui Wei Zhao, Jianguo Li, Yurong Chen, Jia Ming Liu, Yu Gang Jiang, and Xiangyang Xue. Regional Gating Neural Networks for Multi-label Image Classification. *British Machine Vision Conference 2016, BMVC 2016*, 2016-Septe:72.1–72.12, 2016.

[258] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2933–2942, 2017.

[259] Zhenxing Zheng, Gaoyun An, Dapeng Wu, and Qiuqi Ruan. Spatial-temporal pyramid based convolutional neural network for action recognition. *Neurocomputing*, 358:446–455, 2019.

[260] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[261] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *PAMI*, 40(6):1452–1464, 2017.

[262] Howard Zhou, Tucker Hermans, Asmita V Karandikar, and James M Rehg. Movie genre classification via scene categorization. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 747–750, 2010.

[263] Jiaming Zhou, Junwei Liang, Kun-Yu Lin, Jinrui Yang, and Wei-Shi Zheng. Actionhub: A large-scale action video description dataset for zero-shot action recognition. *arXiv preprint arXiv:2401.11654*, 2024.

[264] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[265] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[266] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13516–13525, 2021.

[267] Zixin Zhu, Le Wang, Wei Tang, Ziyi Liu, Nanning Zheng, and Gang Hua. Learning disentangled classification and localization representations for temporal action localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3644–3652, 2022.

[268] Andrew Zisserman, Joao Carreira, Karen Simonyan, Will Kay, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[269] Zoran Zivkovic and Frans van der Heijden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Proceedings of the 17th International Conference on Pattern Recognition*, 2:28–31, 2004.

[270] Mahdi Zolnouri, Xinlin Li, and Vahid Partovi Nia. Importance of data loading pipeline in training deep neural networks. *arXiv preprint arXiv:2005.02130*, 2020.