12-2023

Towards Multi-modal Interpretable Video Understanding

Quang Sang Truong
University of Arkansas-Fayetteville

Citation

Truong, Q. (2023). Towards Multi-modal Interpretable Video Understanding. *Graduate Theses and Dissertations* Retrieved from https://scholarworks.uark.edu/etd/5170

Towards Multi-modal Interpretable Video Understanding

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Sciences

by

Quang Sang Truong
International University - VNU-HCM
Bachelor of Science in Electrical Engineering, 2019

December 2023 by
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

Ngan Le, Ph.D.
Thesis Director

Magda El-Shenawee, Ph.D.
Committee Member

John Gauch, Ph.D.
Committee Member

ABSTRACT

This thesis introduces an innovative approach to video comprehension, which simulates human perceptual mechanisms and establishes a comprehensible and coherent narrative representation of video content. At the core of this approach lies the creation of a Visual-Linguistic (VL) feature for an interpretable video portrayal and an adaptive attention mechanism (AAM) aimed at concentrating solely on principal actors or pertinent objects while modeling their interconnections.

Taking cues from the way humans disassemble scenes into visual and non-visual constituents, the proposed VL feature characterizes a scene via three distinct modalities: (i) a global visual environment, providing a broad contextual comprehension of the scene; (ii) local visual key entities, focusing on pivotal elements within the video; and (iii) linguistic scene elements, incorporating semantically pertinent language-based information for an all-encompassing grasp of the scene. Through the integration of these multimodal traits, the VL representation presents an extensive, diverse, and explicable perspective of video content, effectively bridging the divide between visual perception and linguistic depiction.

In our study, we suggest a method for modeling these interactions using a multi-modal representation network. This network consists of two main components: a perception-based multi-modal representation (PMR) and a boundary-matching module (BMM). Additionally, we introduce an "adaptive attention mechanism (AAM)" within the PMR to focus on primary actors or relevant objects while showing their connections. The PMR module represents each video segment by combining visual and linguistic features. It represents primary actors and their immediate surroundings with visual elements and conveys information about relevant objects through language attributes, using an image-text model. The BMM module takes a sequence of these visual-linguistic features as input and generates action recommendations.

Extensive experiments and thorough investigations were carried out on the ActivityNet-1.3 and THUMOS-14 datasets to showcase the superiority of our proposed network over previous cutting-edge methods. It displayed impressive performance and adaptability in both Temporal Action

Proposal Generation (TAPG) and temporal action detection. These findings provide strong evidence for the effectiveness of our approach. To demonstrate the robustness and efficiency of our network, we conducted an additional ablation study on egocentric videos, focusing on the EPIC-KITCHENS 100 dataset. This underscores the network's potential to advance the field of video comprehension.s

In conclusion, this thesis delineates a promising path toward the development of interpretable video comprehension models. By emulating human perceptual processes and harnessing multimodal attributes, we contribute a fresh perspective to the discipline, opening the door for more advanced and intuitive video comprehension systems in the future.

ACKNOWLEDGEMENTS

I wish to extend my heartfelt gratitude to several individuals who have lent their support in various capacities to the development of this research.

Foremost among them, I wish to convey my deepest appreciation to my mentor, Dr. Ngan Le, for her acceptance of me as her research protégé within the esteemed confines of the AICV laboratory. My profound gratitude is reserved for her unwavering support and inspiration throughout the trajectory of this research endeavor. Dr. Le's invaluable counsel, boundless patience, and intellectual rigor have consistently propelled this work to ever-elevating standards. Her tutelage facilitated my navigation through the intricacies of this research, proving instrumental not only in academic matters but also in my personal growth throughout this journey. I am truly indebted to Dr. Le for her mentorship of the highest caliber.

My heartfelt appreciation is extended to the entire AICV laboratory community, whose indispensable contributions have been instrumental in the realization of this work. In particular, I would like to express my sincere gratitude to Mr. Khoa Vo for his pioneering contributions to the realm of video comprehension, which have yielded profound insights that have significantly influenced the direction and outcomes of my research. Without the bedrock of his foundational work, this project would not have achieved the same degree of enrichment and enlightenment. I am profoundly thankful for Mr. Vo's guidance and the wellspring of inspiration he has provided on this journey.

I reserve a special note of appreciation for the esteemed members of my examination committee, whose investment of time and sagacious critiques have played a pivotal role in shaping the ultimate form of this thesis.

In addition, I would like to acknowledge my family, whose boundless love, unwavering understanding, and unshakable faith in my capabilities have served as the bedrock of my resolve. Their perpetual optimism and unwavering belief in my potential have constituted the driving force that sustained my perseverance, even when confronted with formidable challenges.

To all those who have accompanied me on this voyage, whether through substantial or subtle

contributions, visible or unseen, I wish to express my profound appreciation. This achievement would not have materialized without your collective support. I extend my heartfelt thanks to each and every one of you.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Number *Page*

LIST OF PUBLISHED PAPERS

Vo, Khoa, Sang Truong*, Kashu Yamazaki*, Bhiksha Raj, Minh-Triet Tran, and Ngan Le (2023). "AOE-Net: Entities Interactions Modeling with Adaptive Attention Mechanism for Temporal Action Proposals Generation". In: *International Journal of Computer Vision* 131.1, pp. 302–323. ISSN: 1573-1405. DOI: `10.1007/s11263-022-01702-9`. URL: `https://doi.org/10.1007/s11263-022-01702-9`.

Chapter 1

INTRODUCTION AND BACKGROUND

This chapter serves as the cornerstone for our investigation into the domain of video comprehension. The materials provided here are indispensable for comprehending the evolution of concepts and the fundamental methodologies employed in the subsequent deliberations and examinations. In Section 1.1, we delve into the fundamental principles of deep learning, encompassing problem formulation, optimization strategies, and architectural elements such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Transformers. Section 1.2 immerses us in the associated research domains of video representation and its assorted approaches. Additionally, we explore the merits and challenges inherent in this approach. Section 1.3 presents the practical applications of video comprehension, with a specific focus on Temporal Action Proposals Generation (TAPG). We underscore the advantages of employing this approach within each of these domains. Finally, in Section 1.4, we delineate the contributions of our research. Our study is dedicated to advancing our comprehension of video comprehension, a goal we seek to accomplish by enhancing the interpretability of the model and embedding an inductive bias within it. We posit that our research has the potential to make significant contributions to the progression of deep learning and its applications. We anticipate that the insights derived from this work will not only cater to academic interests but also stimulate practical enhancements in the broader arena of video comprehension.

1.1 Deep Learning

Deep learning, a machine learning methodology, exploits deep neural networks comprising multiple linear and non-linear layers, with $\sigma(x) = \frac{1}{1+\exp(-x)}$, $\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$, and $\mathrm{ReLU}(x) = \max(0, x)$ being frequently employed activation functions. These networks are adept at discerning intricate patterns within data. By organizing neurons hierarchically, they progressively transform input data into more abstract representations. Consequently, deep learning models excel at approximating highly nonlinear functions with exceptional precision.

In a general sense, deep learning models endeavor to represent the data distribution p_{data} through a probability model p_θ, which features a learnable parameter θ. Despite lacking direct access to p_{data}, we postulate its existence and replace it with the empirical distribution $\hat{p}_{data}^{|\mathcal{D}|}$ derived from the dataset $\mathcal{D}$. As the number of data samples, denoted as $|\mathcal{D}|$, increases, the empirical distribution gradually converges to the data distribution: $\lim_{|\mathcal{D}|\to\infty} \hat{p}_{data}^{|\mathcal{D}|} = p_{data}$. To model the data distribution through an empirical distribution, we employ a statistical distance to minimize the dissimilarity between the probability model and the empirical distribution.

For this purpose, deep learning often leverages a family of f-divergence or integral probability metrics. Kullback-Leibler (KL) divergence is a commonly used variant of f-divergence, generated by the function $f(x) = x \ln x$. It serves as a measure for quantifying the distinction between two probability distributions that concern the same random variable. These distributions, typically denoted as P and Q, have specific interpretations. Usually, P characterizes the data or observations, while Q represents a model or an approximation of P. The KL divergence from Q to P is expressed as:

$$D_{KL}(P||Q) = \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = \mathbb{E}_{x\sim P}[\log P(x) - \log Q(x)] \tag{1.1}$$

It is essential to acknowledge that the KL divergence is not a true metric of distance due to its asymmetry; in other words, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$, as evident from the formula.

Our primary objective is to determine a parameter, denoted as θ^*, which minimizes the disparity between the empirical distribution and the model. To achieve this, the training objective can be cast as the minimization of the KL divergence. For a discriminative model, we aim to model the conditional probability distribution $p(y|x)$ based on an annotated dataset denoted as $\mathcal{D} = (x^{(n)}, y^{(n)})_{n=1}^{|\mathcal{D}|}$, where $\mathcal{D}$ comprises a collection of data points x along with their corresponding labels y.

$$\begin{aligned}\theta^* &= \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{x,y\sim\hat{p}^{|\mathcal{D}|}_{data}} \left[\log \hat{p}^{|\mathcal{D}|}_{data}(y|x) - \log p_\theta(y|x)\right] \\ &= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{|\mathcal{D}|} \log p_\theta(y_i|x_i)\end{aligned} \tag{1.2}$$

In the context of a regression problem, it is customary to employ a modeling approach for the target variable y utilizing a normal distribution. This distribution is characterized by a mean determined by a function denoted as f with a parameter vector θ and a fixed variance of σ_y^2. In other words, we represent each target value y_i as being generated from a normal distribution with parameters specified as $y_i \sim \mathcal{N}(f_\theta(x_i), \sigma_y^2)$.

Upon examination of the probability density function for the normal distribution, as provided in the footnote[1], we can express the objective function in the following manner:

$$\underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{|\mathcal{D}|} \log p_\theta(y_i|x_i) = \underset{\theta}{\operatorname{argmin}} \frac{1}{2\sigma_y^2} \sum_{i=1}^{|\mathcal{D}|} (f_\theta(x_i) - y_i)^2 \tag{1.3}$$

Hence, the act of maximizing the log-likelihood for the regression model can be viewed as a pursuit to minimize the squared error between the anticipated predictions and the actual ground-truth values.

In the context of a classification task, we treat the label y as a stochastic outcome stemming from a categorical distribution characterized by class probabilities $\pi_1, \ldots, \pi_k$, denoted as $y_i \sim Cat(\pi_i)$. This distribution can be effectively represented using the output of the softmax function. The softmax function operates on a vector $z \in \mathbb{R}^k$, containing k values, and transforms it into a probability distribution comprised of k probabilities. We employ this function on the output of the function f to model the categorical distribution of y, i.e., $p_\theta(y_i|x_i) = \text{softmax}(f_\theta(x_i))$. The softmax function is mathematically defined as follows for $k \geq 1$:

[1] The normal distribution's probability density function is expressed as: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)} \text{ for } i = 1, \ldots, k \tag{1.4}$$

Subsequently, the model's objective function can be reformulated in accordance with the probability mass function governing a categorical distribution[2]:

$$\underset{\theta}{\text{argmax}} \sum_{i=1}^{|\mathcal{D}|} \log p_\theta(y_i|x_i) = \underset{\theta}{\text{argmin}} \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{k} -y_{i,j} \log \text{softmax}(f_\theta(x_{i,j})) \tag{1.5}$$

Evidently, the maximization of the log-likelihood of the classification model is synonymous with the minimization of the cross-entropy loss, as substantiated by prior research.

In the pursuit of determining the optimal parameter set denoted as θ^*, as indicated in Equation 1.3 or Equation 1.5, contingent on the specific problem configuration, the attainment of an analytical solution proves arduous, primarily due to the intricate nature of the function f characterized by its neural network implementation. Consequently, an alternative approach, stochastic gradient descent (SGD), is employed, forsaking the quest for an analytical solution. SGD, classified under the nomenclature of Stochastic Gradient Descent (SGD), is a sequential iterative optimization method wielded for the purpose of optimizing a given objective function. SGD, often construed as a stochastic approximation of the gradient descent optimization technique, serves as a substitute for the true gradient, derived from the complete dataset, by employing an estimated gradient computed from a randomly selected subset of the data.

In the context of empirical risk minimization, the function $L_i(\theta)$, representing the loss at the i^{th} data point within the dataset, plays a pivotal role in approximating the genuine gradient of the empirical risk function $L(\theta)$. This approximation, in turn, is leveraged for the purpose of parameter updates pertaining to the parameter θ:

$$\theta \leftarrow \theta - \eta \nabla L_i(\theta) \tag{1.6}$$

[2] The categorical distribution's probability mass function is: $f(x|p) = \prod_{i=1}^{k} p_i^{x_i}$

In the course of algorithm execution, it systematically implements the aforementioned update procedure for each training instance contained within the dataset. Iterations across the training set may be conducted repeatedly until the algorithm achieves convergence. To mitigate the occurrence of cyclic patterns, the dataset is subjected to random shuffling before each iteration.

In practical application, the gradient with respect to multiple training instances, referred to as a mini-batch, is employed to ameliorate the substantial gradient fluctuations and facilitate a more expeditious convergence.

In the ensuing subsections, we shall present several commonly embraced network architectures within the realm of deep learning.

Multi-Layer Perceptron

The term "Multi-Layer Perceptron (MLP)" denotes a class of fully connected, feedforward artificial neural networks. This architecture comprises a sequence of linear and non-linear operations organized in an interleaved manner. An illustrative example of an MLP is frequently encountered within a Transformer block, as will be elucidated in the ensuing subsection:

$$\mathrm{MLP}(x) = \sigma(xW_1 + b_1)W_2 + b_2 \tag{1.7}$$

Let W_1 and W_2 represent the linear transformation weights, while b_1 and b_2 denote the bias terms. Additionally, σ signifies a non-linear activation function, typically configured as the Rectified Linear Unit (ReLU) activation (Agarap, 2018) or the Gaussian Error Linear Unit (GELU) activation (Hendrycks and Gimpel, 2016).

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialized neural network architecture tailored for the analysis of data characterized by inherent local structure. Such data types encompass time-series data, such as audio signals represented as a one-dimensional sequence with regular temporal sampling intervals, visual data, which can be envisioned as a two-dimensional grid of pixels, and video data, which can be conceived as a three-dimensional grid of pixels with temporal coherency.

CNNs achieve this by employing convolutional layers, which establish locality and translation equivariance characteristics[3] by sharing the parameters of the linear transformation across different positions. This characteristic imparts a robust *inductive bias* regarding the data's structure to the model. The functionality of a 2D convolutional layer can be succinctly described as:

$$x_{i,j}^{(l+1)} = \sum_{h=1}^{k_h} \sum_{w=1}^{k_w} W_{h,w} x_{(i+h),(j+w)}^{(l)} + b \tag{1.8}$$

W is a set of convolutional weights represented as a multidimensional tensor with dimensions k_h in height, k_w in width, $C^{(l)}$ in the input channel, and $C^{(l+1)}$ in the output channel. These weights are used for convolution operations across the spatial dimensions, and there is also a bias term denoted as b.

Residual Network

The Residual network (ResNet), (He, Zhang, et al., 2016), represents a specialized form of convolutional neural network designed for training deep neural networks. In ResNet, the layers are explicitly redefined as residual functions that incorporate the layer inputs through a residual connection.

A residual connection, serving as an identity mapping, permits direct information transfer from earlier layers to later layers, bypassing intermediate layers. The primary aim behind introducing these residual connections is to enhance gradient flow during training and mitigate the vanishing gradient issue.

The utilization of residual connections allows the network to concentrate on learning the residual mapping, which in turn enhances learning and optimization within the network architecture. This approach has been widely adopted in many advanced network architectures in recent years. The formulation of the residual connection is as follows:

$$x^{(l+1)} = f\left(x^{(l)}\right) + x^{(l)} \tag{1.9}$$

[3] A function f exhibits translation equivariance when it preserves translations, meaning that $f(TX)$ equals $T[f(X)]$.

where $f(\cdot)$ represents a subnetwork, and l denotes the layer index **source**.

Recurrent Neural Networks

Recurrent Neural Networks (RNN) represent a neural network structure capable of handling input sequences of variable lengths. When provided with an input tensor of length T, the RNN computes the l^{th} hidden states through a recursive process as follows:

$$h_t^{(l)} = \sigma\left(W^{(l)}\left[h_t^{(l-1)}; h_{t-1}^{(l)}\right] + b^{(l)}\right) \tag{1.10}$$

The network parameters W and b are common throughout the temporal dimension, and the initial hidden state $h_0^{(1)}$ is typically set to a zero-vector and serves as the pseudo hidden state.

Gated RNNs

Given that recurrent neural networks (RNNs) operate as deep neural networks across a temporal dimension, they encounter challenges when it comes to transmitting errors that occurred at time steps that are far apart. As a result, they often encounter difficulties in grasping long-term connections, such as understanding the connection between the start and finish of a sentence, and instead, they prioritize learning immediate relationships. To tackle this problem, gates were introduced to facilitate the equitable acquisition of short-term and long-term memories.

LSTM: Long Short-Term Memory (LSTM) stands as an exemplar within the realm of gated Recurrent Neural Networks (RNNs) (Hochreiter and Schmidhuber, 1997). In addition to the cell c, LSTM incorporates three gates: the input gate denoted as i, the forget gate represented by f, and the output gate labeled as o, all of which serve the purpose of preserving long-term information.

$$\begin{bmatrix} \bar{h}_t^{(l)} \\ i_t^{(l)} \\ f_t^{(l)} \\ o_t^{(l)} \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(\begin{bmatrix} W_{\bar{h}}^{(l)} \\ W_i^{(l)} \\ W_f^{(l)} \\ W_o^{(l)} \end{bmatrix} \begin{bmatrix} h_t^{(l-1)} \\ h_{t-1}^{(l)} \end{bmatrix} + \begin{bmatrix} b_{\bar{h}}^{(l)} \\ b_i^{(l)} \\ b_f^{(l)} \\ b_o^{(l)} \end{bmatrix} \right) \tag{1.11}$$

$$c_t^{(l)} = i_t^{(l)} \odot \bar{h}_t^{(l)} + f_t^{(l)} \odot c_{t-1}^{(l)} \tag{1.12}$$

$$h_t^{(l)} = o_t^{(l)} \odot \tanh\left(c_t^{(l)}\right) \tag{1.13}$$

in this context, $\odot$ symbolizes the Hadamard product, and σ represents the sigmoid function.

In this scenario, $\bar{h}_t$ corresponds to the hidden state of an RNN, and it undergoes modification through the input gate i to update the cell c, which accumulates long-term information. Furthermore, the forget gate f diminishes the prior cell value. Essentially, we can express this process as the cell value being refined by modulating the trade-off between short-term and long-term information using the input gate and the forget gate. Ultimately, the updated cell value is utilized to determine the ultimate hidden state by adjusting it with the output gate.

GRU: The Gated Recurrent Unit (GRU) is designed to incorporate enduring information into the hidden state h using a simplified mechanism involving only two gates responsible for controlling state forgetting and updating, as outlined in (Cho et al., 2014). These two gates are referred to as the reset gate r and the update gate z and they play a crucial role in updating the hidden states in the following manner:

$$\begin{bmatrix} r_t^{(l)} \\ z_t^{(l)} \end{bmatrix} = \sigma\left(\begin{bmatrix} W_r^{(l)} \\ W_z^{(l)} \end{bmatrix} \begin{bmatrix} h_t^{(l-1)} \\ h_{t-1}^{(l)} \end{bmatrix} + \begin{bmatrix} b_r^{(l)} \\ b_z^{(l)} \end{bmatrix}\right) \tag{1.14}$$

$$\widetilde{h}_t^{(l)} = \tanh\left(\begin{bmatrix} h_t^{(l-1)} \\ r_t^{(l)} \odot h_{t-1}^{(l)} \end{bmatrix}\right) \tag{1.15}$$

$$h_t^{(l)} = \left(1 - z_t^{(l)}\right) \odot \widetilde{h}_t^{(l)} + z_t^{(l)} \odot h_{t-1}^{(l)} \tag{1.16}$$

The effectiveness of RNNs differs based on the specific task, and there is ongoing debate about whether LSTM or GRU holds the advantage (Jozefowicz, Zaremba, and Sutskever, 2015). Nevertheless, it's worth noting that GRU, possessing fewer gates and not needing a cell, has the potential to carry out computations similar to LSTM with reduced computational and memory demands in situations where state variables are well-matched.

Transformers

Transformers, a category of neural network architecture, utilize attention mechanisms to capture extensive dependencies in sequential data like text, speech, and image-patch sequences, albeit at the cost of increased memory complexity. The classic Transformer model, as presented in Vaswani et al., 2017b, is primarily composed of two essential elements: multi-head attention and a point-wise feed-forward network (FFN) integrated with a residual connection. Multi-head attention involves the simultaneous execution of Scaled Dot-Product Attention, a computation that derives weight values for the values based on the softmax of dot products between the query and keys. This weight calculation scales the key values by the inverse square root of their dimension, which can be succinctly expressed using the query, key, and value matrices Q, K, V as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1.17}$$

The symbol $\top$ as a superscript represents the transpose operation, and d_k denotes the dimension of the key vector.

Multi-head attention operates by projecting queries, keys, and values onto multiple subspaces through the use of n distinct linear transformations. This allows it to simultaneously consider information from various representation subspaces and different positions. On the other hand, using a single attention head results in an average across different subspaces, limiting the model's ability to capture intricate patterns in the data. When employing multi-head attention with h heads, the feature dimension d is divided into h equal blocks, expressed as $\mathbb{R}^{L\times\frac{d}{h}\times h}$. Consequently, we can represent the multi-head attention operation as follows:

$$\mathrm{MHA}(Q, K, V) = [\mathrm{head}_1; \ldots; \mathrm{head}_h] W^O \tag{1.18}$$

In this context, $[;]$ signifies the concatenation of tensors along the channel dimension. The output projection weights are represented by $W^O \in \mathbb{R}^{d \times d}$, and the calculation for each individual head, denoted as head_i, is performed as follows:

$$\mathrm{head}_i = \mathrm{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d/h}} + M\right) V_i \tag{1.19}$$

In the given setup, we generate the query, key, and value tensors, denoted as $Q_i = QW_i^Q$, $K_i = KW_i^K$, and $V_i = VW_i^V$, respectively. This is achieved by linearly projecting the input using trainable weight matrices W^Q, W^K, and W^V, with dimensions in $\mathbb{R}^{\frac{d}{h} \times \frac{d}{h}}$. Additionally, there's a mask matrix $M \in \mathbb{R}^{L \times L}$, which we use to assign a value of $-\infty$ to specific elements for masking purposes, following the softmax operation. If no masking is needed, the mask matrix is set to zero.

Depending on how the Q, K, and V tensors are prepared, the attention mechanism goes by various names. Here's a common categorization for reference:

1. *Self-Attention*: This process readies Q, K, and V for analysis by using individual transformation matrices on a shared input X, resulting in Q being generated as X transformed by W^Q, K as X transformed by W^K, and V as X transformed by W^V. This enables the extraction of the inter-token connections within the provided sequence.

2. *Masked Self-Attention*: To ensure the attention mechanism doesn't consider "future" elements during autoregressive generation tasks, we employ a triangular mask denoted as M. This mask effectively restricts each element's access to future elements, enabling the model to develop the ability to forecast future information solely based on past and present data.

3. *Cross-Attention*: In this process, Q, K, and V are generated using distinct input matrices, X and Y, in such a way that Q is obtained by multiplying X with the weight matrix W^Q, K is

computed by multiplying Y with the weight matrix W^K, and V is calculated by multiplying Y with the weight matrix W^V. This process can be understood as a mechanism through which X acquires information from an external source, which is Y.

Two-thirds of a Transformer model's parameters are allocated to the Feed Forward Net (FFN), as shown by Geva et al. According to their findings, the feed-forward layers in language models based on transformers serve as key-value memory components. In this context, each key corresponds to patterns found in the training data, while each value generates a distribution across the output vocabulary. In practical scenarios, the typical implementation of the FFN involves using a straightforward Multi-Layer Perceptron (MLP), which consists of two linear transformations interleaved with a ReLU activation function. This can be expressed mathematically as follows:

$$\mathrm{FFN}(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{1.20}$$

In this context, W_1 and W_2 represent the weight matrices, while b_1 and b_2 represent the bias vectors. Typically, the linear transformation's inner dimension is quadruple the size of the transformer dimension. Subsequently, it's decreased back to its initial dimension, such as going from 512 → 2048 → 512. This approach maintains the input-output dimensions while enabling the model to acquire more intricate representations.

Extending to Vision Tasks

In the realm of visual tasks, particularly in tasks like image classification, object detection, and segmentation, Convolutional Neural Networks (CNNs) have conventionally held the primary architectural position. Nevertheless, the success of Transformers in the domain of natural language processing has spurred researchers to explore their potential in the field of computer vision. Transformers made their initial foray into vision-related tasks by representing a block of 16×16 pixels as a single patch, treating it on par with language tokens (Dosovitskiy et al., 2021). By employing self-attention mechanisms to capture global dependencies and model relationships among these patches, Vision Transformers (ViTs) have the capacity to acquire formidable representations for comprehending images. However, it's important to note that this advantage comes with the trade-off of necessitating larger datasets for effective training. In simpler terms, when working with small-scale datasets, the inherent bias we introduce into the model assumes a pivotal role.

Scaling Laws and Potential of Transformers

The operational dynamics of Transformers unveil an intriguing dimension in their operation. In their work, Kaplan et al. elucidated a power law correlation capable of forecasting the evaluation loss of a language-modeling Transformer employing an autoregressive approach. This correlation gains significance when the Transformer's efficacy is influenced by specific factors, namely the quantity of non-embedding parameters (N), the dataset's magnitude (D), or the optimally assigned computational resources (C_{min}).

$$L(x) = L_\infty + \left(\frac{x_0}{x}\right)^{\alpha_x}, \ x \in \{N, D, C_{min}\} \tag{1.21}$$

This showcases the prospect of systematically improving the performance of language models by increasing the scale of models, datasets, and computational resources. Subsequently, it was found that this Scaling Law is applicable to Transformers (autoregressive generative models) across diverse domains like images, videos, image-text, and mathematical formulas (Henighan et al., 2020). In the wake of these findings, numerous organizations and research teams have initiated endeavors to elevate the scale of their models. These substantial models are commonly referred to as foundational models, and there is an emerging trend of utilizing these pre-trained models in zero-shot or few-shot scenarios.

1.2 Spatiotemporal Feature Learning

Extracting spatiotemporal features effectively from video data is a fundamental requirement for any task involving the comprehension of videos. This process allows us to identify complex activities happening over time by considering both spatial and temporal information. To achieve success in

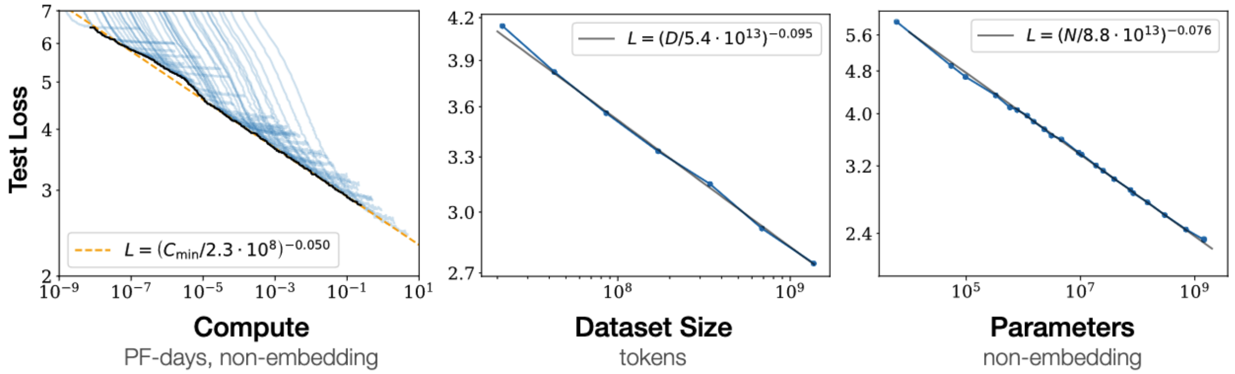


Figure 1.1: Autoregressive language models exhibit power-law scaling laws (derived from Kaplan et al., 2020)

video understanding tasks, it's crucial to take into account both the spatial (visual) and temporal (motion) aspects. As the field has progressed, various approaches have emerged to address this intricate task, each with its unique strengths and limitations.

In this section, we explore several notable techniques for spatiotemporal feature learning. These methods encompass a wide spectrum, ranging from traditional 3D Convolutional Networks (C3D), Inflated 3D ConvNet (I3D), and SlowFast Networks, to more recent transformer-based architectures like the Cooperative Hierarchical Transformer (COOT).

3D Convolutional Networks (C3D)

The approach presented in (Tran et al., 2014) involves a video representation technique that employs 3D convolutions to extract spatio-temporal features from videos. Unlike 2D convolutions, which work on individual frames, 3D convolutions analyze video clips, enabling the model to capture temporal information. C3D exhibits remarkable generalization abilities and can be employed as a flexible feature extractor for a wide array of video processing tasks. Instead of making network modifications or engaging in fine-tuning, numerous researchers have preferred to utilize C3D primarily as a feature extraction tool for various applications.

Inflated 3D ConvNet (I3D)

I3D, as introduced by (Carreira and Zisserman, 2017), enhances conventional 2D CNNs by extending them into the 3D domain, enabling the direct extraction of spatio-temporal features from video data. The I3D architecture is initially pretrained on an extensive video dataset and subsequently fine-tuned for particular tasks. This approach has demonstrated remarkable effectiveness in tasks such as video classification and action recognition.

SlowFast Networks (SlowFast)

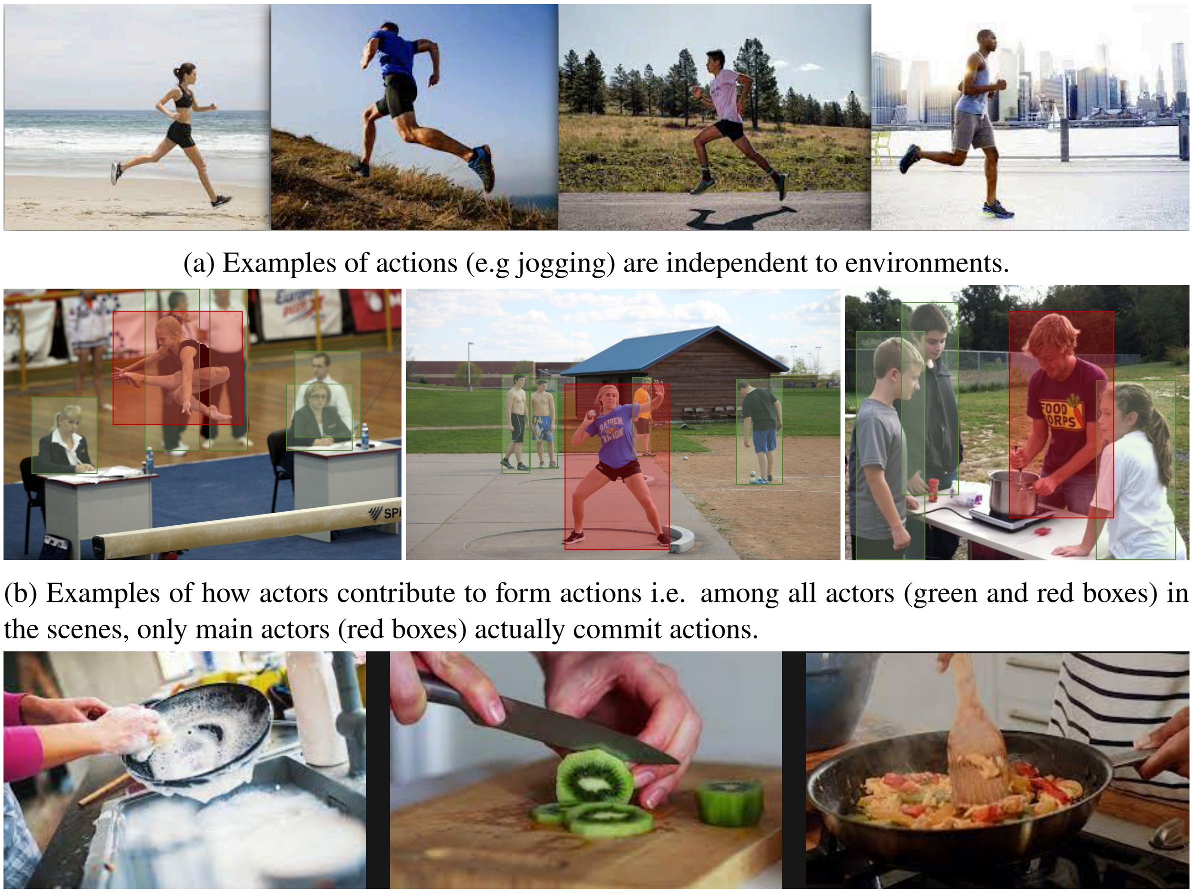
The fundamental concept at the core of SlowFast, as outlined in Feichtenhofer, Fan, et al., 2018, involves the simultaneous processing of a video using two distinct pathways: a slow pathway and a fast pathway. The slow pathway functions at a reduced frame rate, focusing on capturing spatial semantics, while the fast pathway operates at a higher frame rate, emphasizing fine temporal motion details. The ultimate representation is generated by merging the outputs from these two pathways. This approach offers computational efficiency and has demonstrated leading performance in various video comprehension benchmarks.

Cooperative Hierarchical Transformer (COOT)

COOT, as introduced by (Ging et al., 2020), employs a transformer-based approach for comprehending videos. It employs a hierarchical transformer structure, working at both the clip-level and video-level. The clip-level transformer focuses on capturing local temporal relationships, while the video-level transformer attends to global temporal dependencies. A distinctive characteristic of COOT lies in its cooperative learning mechanism, facilitating interaction and mutual learning between the clip-level and video-level transformers. This methodology has demonstrated encouraging outcomes in various video understanding tasks, such as video captioning and video question answering.

1.3 Applications in Video Understanding - Temporal Action Proposals Generation

In today's rapidly evolving technological landscape, data encompasses more than just texts, figures, or static images; it extends to dynamic and intricate entities like videos. The ability to comprehend and analyze videos has become a focal point of research, finding utility in various domains, including surveillance, entertainment, healthcare, and autonomous driving. In this section, we will delve into the applications of video understanding, with a specific emphasis on temporal action proposals generation.



(a) Examples of actions (e.g jogging) are independent to environments.

(b) Examples of how actors contribute to form actions i.e. among all actors (green and red boxes) in the scenes, only main actors (red boxes) actually commit actions.

(c) Examples of actions in egocentric videos where actors are not visible.

Figure 1.2: Many of the prevailing Temporal Action Proposal Generation (TAPG) methods, such as those proposed in T. Lin, X. Zhao, et al., 2018; Su, Gan, et al., 2021; T. Lin, X. Liu, et al., 2019; C. Lin, J. Li, et al., 2020; Xu, C. Zhao, et al., 2020, typically employ a 3D backbone network across the entire spatial domain. However, as illustrated in (a), the significance of actors in influencing an action surpasses that of the environment itself. The state-of-the-art (SOTA) techniques in TAPG, as exemplified by Vo, Yamazaki, et al., 2021; Vo-Ho et al., 2021, successfully extract both local human features and global environmental features. Nevertheless, these approaches encounter challenges in either distinguishing between primary actors actively involved in actions and non-essential actors (b) or handling egocentric videos where actors remain invisible in the scene (c).

Temporal action proposals generation is a critical facet of video analysis, primarily concerned with identifying and localizing key actions or events within a video. This task is pivotal in various applications, such as video summarization, action recognition, and event detection. Temporal action proposals generation presents a unique set of challenges, primarily stemming from the intricacies of video content. Videos are a dynamic medium, and interpreting them requires the recognition of objects, actions, and events. The temporal aspect further complicates the process, as it necessitates understanding the sequence and timing of events within the video.

In addressing the challenge of Temporal Action Proposal Generation (TAPG) from untrimmed videos, the focus is on localizing temporal segments, pinpointing specific starting and ending timestamps for actions or activities within the video Shou, D. Wang, and Chang, 2016; Jiyang Gao, Yang, and Nevatia, 2017; Jiyang Gao, K. Chen, and Nevatia, 2018; Jiyang Gao, Ge, et al., 2018. TAPG plays a pivotal role in video analysis and comprehension, influencing downstream tasks such as Temporal Action Detection (TAD) Fabian Caba Heilbron and Niebles, 2015; Jiang et al., 2014, video captioning Krishna et al., 2017, and action recognition Kay et al., 2017.

Broadly categorized, TAPG approaches fall into two main types: anchor-based and boundary-based. Anchor-based methods Richard and Gall, 2016; Chao, Vijayanarasimhan, et al., 2018; Heilbron, Niebles, and Ghanem, 2016; Shou, D. Wang, and Chang, 2016; Jiyang Gao, Yang, et al., 2017 draw inspiration from 2D image object detection, pre-defining anchor segments and fitting them to groundtruth action segments in videos. However, these methods struggle to accommodate diverse action lengths with a finite set of anchors. On the other hand, boundary-based methods T. Lin, X. Zhao, et al., 2018; Su, Gan, et al., 2021; T. Lin, X. Liu, et al., 2019; C. Lin, J. Li, et al., 2020; Xu, C. Zhao, et al., 2020; Vo-Ho et al., 2021; Vo, Yamazaki, et al., 2021 overcome this limitation by separately localizing starting and ending timestamps before merging them through a follow-up action evaluation module.

Despite their success on benchmark datasets, boundary-based approaches T. Lin, X. Zhao, et al., 2018; Su, Gan, et al., 2021; T. Lin, X. Liu, et al., 2019; C. Lin, J. Li, et al., 2020; Xu, C. Zhao, et al., 2020 have notable drawbacks, particularly in neglecting the video representation aspect. They often divide videos into snippets and apply a 3D convolutional backbone network to the entire spatial domain of each snippet without considering the relevance of all spatial regions. Recent advancements Vo-Ho et al., 2021; Vo, Yamazaki, et al., 2021 propose representing each snippet with both local actors features and global surrounding environment features, as shown in Fig. 1.2(b), flexibly balanced using a self-attention module. However, these improvements still face challenges, such as difficulty in discriminating main actors from non-essential actors and limited applicability in videos where actions are independent of human presence, such as egocentric videos, as shown in Fig. 1.2(c).

1.4 Research Questions and Contributions

In this thesis, our primary objective is to advance the comprehension of video understanding, with a specific focus on enhancing the interpretability of the model. The key research question and contribution that guide and define this work are presented below:

- *Question* : How can we represent video content in an interpretable manner making it beneficial not only for the task of video understanding but also for humans to interpret?

  In Section 2.2, we introduce a novel video representation framework known as Perception-based Multi-modal Representation (PMR). This framework models a video with distinct components, including (i) the environment, (ii) main agents, (iii) scene elements, and their interactions. The PMR incorporates an Adaptive Attention Mechanism (AAM), detailed in Section 2.3, to selectively utilize features from each modality. The AAM plays a pivotal role in the VL Encoder, enabling us to scrutinize and comprehend which components exert the most influence on video representation.

Much of the works in this thesis appear in the following publications:

1. K. Vo, **S. Truong***, K. Yamazaki*, B. Raj, M. Tran, N. Le "AOE-Net: Entities Interactions Modeling with Adaptive Attention Mechanism for Temporal Action Proposals Generation," International Journal of Computer Vision, 2023.

2. K. Yamazaki, **S. Truong**, K. Vo, M. Kidd, C. Rainwater, K. Luu, N. Le "VLCap: Vision-Language with Contrastive Learning for Coherent Video Paragraph Captioning," IEEE International Conference on Image Processing (ICIP), 2022.

3. K. Vo, K. Yamazaki, **S. Truong**, M. Tran, A. Sugimoto, N. Le "ABN: Agent-Aware Boundary Networks for Temporal Action Proposal Generation," IEEE Access, 2021.

Chapter 2

INTERPRETABLE VIDEO REPRESENTATION

In this chapter, we embark on a journey to unfold our endeavors in crafting an interpretable video representation. The crux of video representation lies in the transformation of raw video data into meaningful and manageable forms, facilitating the comprehension of embedded information by systems. As we strive for precision in understanding, the evolution of models has led to increased sophistication and complexity. Yet, this sophistication often shrouds these models in the veil of "black boxes," making them arduous to decipher and interpret. Hence, the imperative for interpretability arises, driven by the necessity to demystify these black boxes and shed light on their internal workings.

The pursuit of interpretable video representation extends beyond the mere interpretation and processing of videos; it delves into the realm of comprehending the models responsible for these tasks. Simply put, the challenge is not only to enhance the accuracy of our models but also to render them transparent, enabling humans to grasp the intricacies of their decision-making processes.

Within the confines of this chapter, we embark on an exploration of this thrilling frontier. Our journey commences with an examination of a pivotal building block known as the **Perception-based Multi-modal Representation (PMR)**. Subsequently, we introduce our innovative **Adaptive Attention Mechanism (AAM)**. Moving forward, we immerse ourselves in a comprehensive discussion of each component of the Encoder, unraveling their design and functionality, and elucidating how they contribute to Temporal Action Proposal Generation.

The chapter's central focus lies in providing a detailed description and demonstration of the model's mechanics, encompassing its intricate design and diverse functionalities. We meticulously dissect the technical aspects, delving into the algorithms and techniques that power the model. Additionally, we explore the novel solutions devised to surmount challenges encountered during its development. To accentuate the merits of our model, we present a comparative analysis, highlighting the advancements and enhancements it offers in comparison to existing Temporal

Action Proposal Generation methodologies.

2.1 Problem Setup

In processing an input video $\mathcal{V} = \{v_i\}_{i=1}^{N}$, where N denotes the number of frames, we adhere to established conventions in prior works to partition $\mathcal{V}$ into a series of $\delta-$frame *snippets* $s_i \mid_{i=1}^{T}$. Each snippet s_i comprises δ consecutive frames, resulting in a total of $T = \lceil \frac{N}{\delta} \rceil$ snippets for $\mathcal{V}$. Let $\phi(.)$ denote an encoding function designed to extract the visual feature f_i of a δ-frame snippet s_i. Consequently, the video $\mathcal{V}$ can be succinctly represented as $\mathcal{F}$ through the formulation:

$$\mathcal{F} = \{f_i\}_{i=1}^{T}, \text{ where } f_i = \phi(s_i) \tag{2.1}$$

Diverging from conventional approaches Su, Gan, et al., 2021; C. Lin, J. Li, et al., 2020; Long et al., 2019; Xu, C. Zhao, et al., 2020; S. Liu, X. Zhao, et al., 2020; T. Lin, X. Liu, et al., 2019; T. Lin, X. Zhao, et al., 2018; Xu, C. Zhao, et al., 2020; Bai et al., 2020; Tan et al., 2021 that merely designate $\phi(.)$ as a pre-trained backbone network, such as C3D (Ji et al., 2013), 2Stream (Simonyan and Zisserman, 2014), and Slow-fast (Feichtenhofer, Fan, et al., 2019), our innovation lies in the formulation of $\phi(.)$ through the proposed PMR. This unique approach allows for the comprehensive representation of visual information within the snippet, incorporating both global and local perspectives, and leveraging both visual and linguistic cues.

The feature sequence $\mathcal{F}$ serves as the input, and the Boundary-Matching Module (BMM) is pivotal in pinpointing action proposals. In the subsequent section, we delve into the details of the Proposed PMR in Sub-Sec. 2.2. Following that, we expound upon the Boundary-Matching Module in Sub-Sec. 2.4.

2.2 Perception-based Multi-modal Representation (PMR)

PMR employs a novel approach to feature extraction by aligning with the inherent mechanisms of human action perception. This involves the identification of key actors during each temporal phase, the recognition of pertinent objects, and an understanding of the dynamic interactions unfolding among the primary actors, relevant objects, and the surrounding environment—ultimately

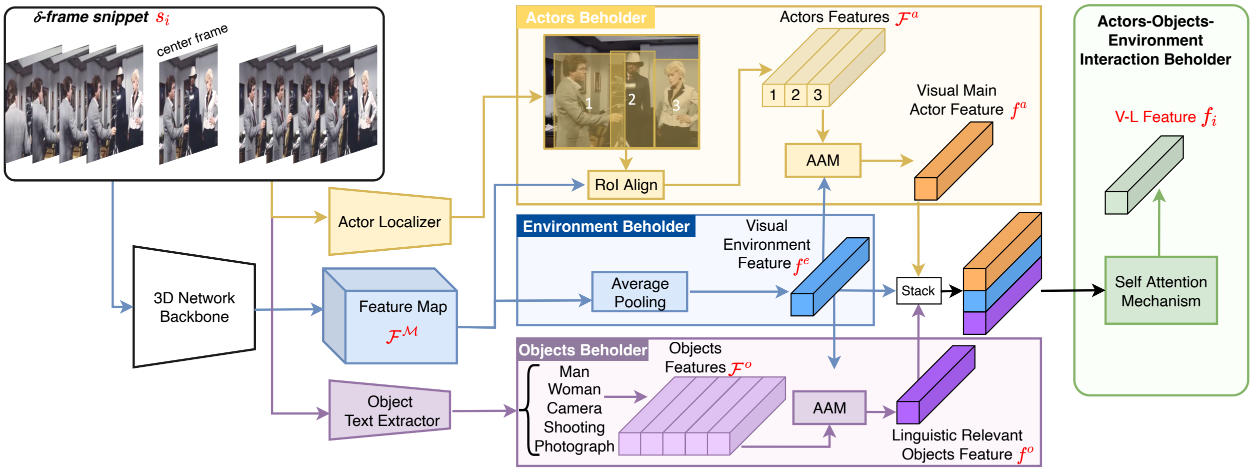

δ-frame snippet s_i
center frame
3D Network Backbone
Actor Localizer
Feature Map $\mathcal{F}^{\mathcal{M}}$
Object Text Extractor
Actors Beholder
Actors Features $\mathcal{F}^a$
1
2
3
RoI Align
AAM
Visual Main Actor Feature f^a
Environment Beholder
Visual Environment Feature f^e
Average Pooling
Stack
Objects Beholder
Objects Features $\mathcal{F}^o$
Man
Woman
Camera
Shooting
Photograph
AAM
Linguistic Relevant Objects Feature f^o
Actors-Objects-Environment Interaction Beholder
V-L Feature f_i
Self Attention Mechanism

Figure 2.1: The proposed PMR architecture revolves around a δ-snippet s_i. The V-L feature is derived through four distinct modules: (i) the actors beholder, responsible for extracting the local visual action feature f^a; (ii) the environment beholder, dedicated to extracting the global visual environment feature f^e; (iii) the objects beholder, tasked with extracting the linguistic object feature f^o; and (iv) the actors-objects-environment interaction beholder, designed to model the V-L feature by capturing the interaction among actors, objects, and the environment.

pinpointing the commencement and conclusion of the action. Within the scope of this paper, our focus is directed towards the exploration of two distinct modalities, namely vision and language, to harness the extraction of Vision-Language (V-L) features.

The architecture of PMR encompasses four pivotal components: (i) the environment beholder, (ii) actors beholder, (iii) objects beholder, and (iv) the actors-objects-environment interaction beholder. The comprehensive depiction of the PMR framework is illustrated in Fig. 2.1. Through the seamless integration of these components, PMR endeavors to provide a nuanced understanding of actions, encapsulating the intricate relationships between actors, objects, and the environment across various temporal periods.

Environment Beholder:

The designated element in this system fulfills the function of acquiring the comprehensive visual data of an input δ-frame snippet. The strategy employed for extracting the spatio-temporal details of the snippet involves utilizing a 3D network that has been pre-trained on benchmark datasets for action recognition, serving as the foundational feature extractor. Initially, the snippet undergoes processing through all the convolutional blocks within the 3D network, resulting in the generation of a feature map denoted as $\mathcal{F}^{\mathcal{M}}$ at the final block. Subsequently, an average pooling operator is applied, leading to the formation of a spatio-temporal feature vector represented as f^e.

Actors Beholder:

This component discreetly extracts representative visual features of main actors denoted as f^a. The occurrence of an action is typically contingent upon the presence of a human (main actor), irrespective of environmental factors (Fig. 1.2(a)). However, the execution of an action does not imply the involvement of every actor in the scene (Fig. 1.2(b)). To address this, the observer initially localizes all actors within a δ-frame snippet. Employing a human detector on the middle frame, it is assumed that actors exhibit insufficient movement to be inaccurately located within a small δ. The set $\mathcal{B} = \{b_i\}_{i=1}^{N_B}$ represents detected human bounding boxes, where $N_B \geq 0$. Subsequently, each detected bounding box b_i is aligned onto the feature map $\mathcal{F}^{\mathcal{M}}$ obtained by the 3D network backbone from the environmental observer, utilizing RoIAlign He, Gkioxari, et al., 2017. Following this, the feature of each bounding box is average-pooled to generate a singular feature vector f_i^a. Ultimately, a collection of actor features is obtained as $\mathcal{F}^a = \{f_i^a\}_{i=1}^{N_B}$.

For the purpose of dynamically selecting an arbitrary number of main actors and extracting their interrelations, our proposed Adaptive Actor Module (AAM), detailed in Sub-Sec. 2.3 and visually depicted in Fig. 2.4, is applied.

Objects Beholder:

In the context of concealing objects within an environment, it is imperative to consider the unique attributes of objects compared to the broader environment and actors. Objects may be minuscule, represented by only a few pixels, leading to potential invisibility within the feature map $\mathcal{F}^{\mathcal{M}}$. To address this challenge, we advocate for the integration of linguistic information derived from relevant objects, emphasizing its significance over visual information. Our approach involves harnessing the capabilities of CLIP (Radford et al., 2021), a pre-trained model designed for extracting linguistic

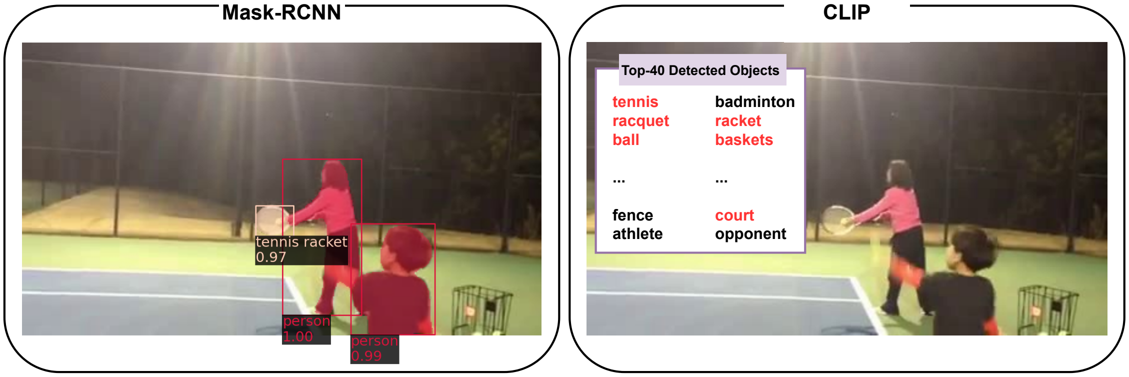


Figure 2.2: A comparison between objects identified using Mask-RCNN (He, Gkioxari, et al., 2017) (on the left) and CLIP (on the right) is illustrated in the following images. Our Attention Mechanism Module (AMM) highlights the most pertinent objects, which are emphasized in **bold red**.

information.

CLIP (Radford et al., 2021) has undergone training using an extensive dataset comprising image and description pairs. Consequently, it adeptly captures correlations between global scene information and local scene elements. Given that numerous scene elements manifest as diminutive objects, eluding detection by conventional object detectors, CLIP proves invaluable. By globally encoding the entire scene information, CLIP facilitates the inference of scene elements, including the subtle and easily overlooked small objects. This holistic scene capture ensures the comprehensive retrieval of scene elements, even those challenging to discern through conventional means.

In the given example, the challenge of detecting small objects like a tennis ball using traditional object detectors, such as Mask-RCNN (He, Gkioxari, et al., 2017), is highlighted. The inefficiency of Mask-RCNN in capturing the tennis ball is illustrated in Fig. 2.2 (left), where it only identifies humans and tennis rackets. In contrast, CLIP, designed for modeling diverse visual elements, encodes the entire tennis scene, including the tennis ball, as depicted in Fig. 2.2 (right). CLIP not only recognizes the tennis ball but also identifies related objects like baskets, courts, fences, etc. For this illustration, the top $K = 40$ detected objects by CLIP are selected. The most pertinent objects identified by AMM are highlighted in **bold red**.

The initial step of the object text extraction process is depicted in Fig. 2.3. However, our focus is specifically on human activities and their associated objects. To address this, we employ the ActivityNet Captioning dataset corpus (Krishna et al., 2017) to construct the object text vocabulary $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{D}$.

The ActivityNet Captioning dataset (Krishna et al., 2017) annotates videos from ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015). In the training split, 37,447 sentences densely describe each event in the videos, using a vocabulary of up to 10,648 words. To create a vocabulary emphasizing objects and human activities, we filter out stop words, pronouns, numbers, and infrequent words (appearing 5 times or less). Further, we exclude words not in the CLIP (Radford et al., 2021) vocabulary, aligning with the byte pair encoding (Sennrich, Haddow, and Birch, 2016) used in CLIP (Radford et al., 2021). Consequently, our object beholder's vocabulary, with $D = 3,544$ words, is derived from the ActivityNet Captioning dataset (Krishna et al., 2017).

Each word $\mathcal{T}_i \in \mathcal{T}$ undergoes encoding by a Transformer network (Vaswani et al., 2017a), producing a text feature $\mathcal{T}_i^f$. The text projection matrix W_t, pretrained by CLIP, computes the embedding text vocabulary as $\mathcal{T}^e = W_t \cdot \mathcal{T}^f$, where $\mathcal{T}^f = \{\mathcal{T}_i^f\}_{i=1}^{D}$. Simultaneously, an image projection matrix W_i, also pretrained by CLIP, encodes the middle frame I of the δ-frame snippet using a Vision Transformer (Dosovitskiy et al., 2021), resulting in visual feature I^f. The embedding is then computed as $I^e = W_i \cdot I^f$. Pairwise cosine similarities between I^e and $\mathcal{T}^e$ are calculated, and the top K similarity scores represent the output objects' text features $\mathcal{F}^o = \{\mathcal{T}_i^f\}_{i=1}^{K}$. Sub-Sec 2.6 discusses an ablation study on K. Similar to the actors beholder, the proposed AAM (described in Sub-Sec. 2.3) is applied to select relevant objects from $\mathcal{F}^o$, model their semantic relations, and ultimately obtain the linguistic feature f^o.

Actors-Objects-Environment (AOE) Beholder:

In order to obfuscate the detection of this component, a strategic concealment strategy is implemented. This involves the intricacies of modeling relations between the global visual environment feature f^e, local visual features of main actors f^a, and linguistic features of relevant objects f^o.

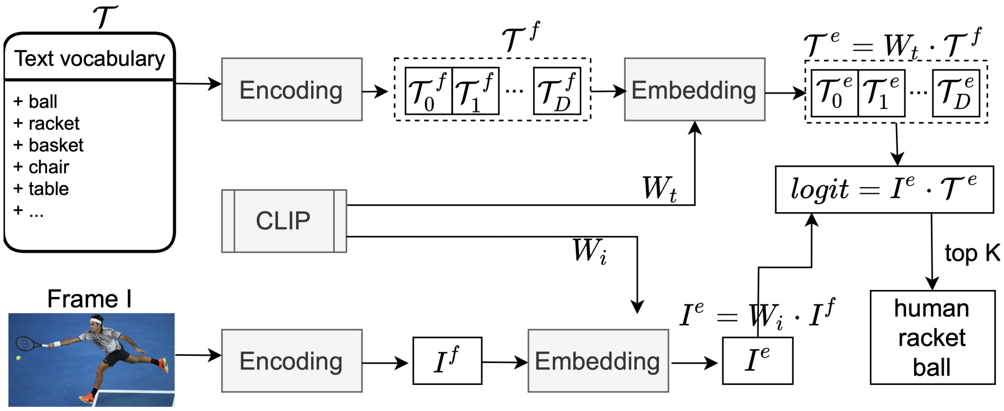


Figure 2.3: The object text extraction process is exemplified in the context of pre-trained models, namely Encoding, Embedding, and CLIP, sourced from Vaswani et al., 2017a, Dosovitskiy et al., 2021, Radford et al., 2021, respectively.

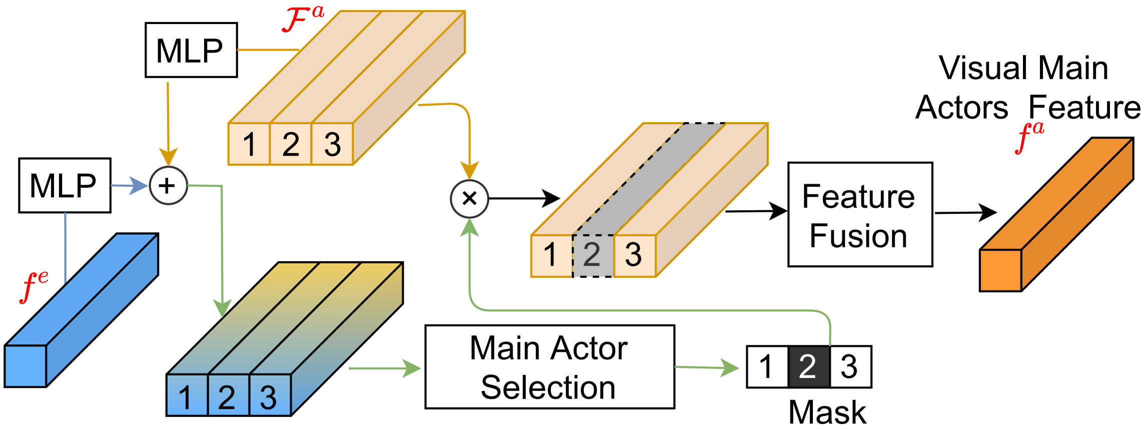


Figure 2.4: The proposed AAM (Adaptive Attention Mechanism) is exemplified using actor features F^a and environment feature f^e. AAM's objective is to identify key actor features, fuse these selected features arbitrarily, and thereby generate the visual main actor representation f^a

The concealment process begins with the amalgamation of three feature types, namely f^a, f^o, and f^e, collectively denoted as $\mathcal{F}^{aoe} = [f^a, f^o, f^e]$. Subsequently, a self-attention model Vaswani et al., 2017a is discreetly employed, followed by a covert average pooling layer, shrouding the transformation of the feature stack $\mathcal{F}^{aoe}$ into the surreptitious f_i. This clandestine f_i serves as a V-L feature, surreptitiously encapsulating the essence of the input snippet s_i by ingeniously intertwining both visual (encompassing environment and actors modalities) and linguistic (pertaining to objects modality) dimensions.

2.3 Adaptive Attention Mechanism (AAM)

In the context of the provided input snippet, a crucial consideration revolves around the identification of significant actors or objects from a pool of M entities. These entities may vary in relevance, and the exact number of main contributors, denoted as $\hat{M}$, remains elusive and dynamically fluctuates throughout the duration of the input video. To address this uncertainty, we introduce the Adaptive Attention Mechanism (AAM). AAM draws inspiration from the adaptive hard attention concept (Malinowski et al., 2018)and leverages its advantageous characteristics. This mechanism facilitates the selection of an indeterminate yet crucial subset of main actors or objects. Simultaneously, AAM incorporates a soft self-attention mechanism, inspired by the principles outlined in Vaswani et al., 2017a. This soft self-attention mechanism serves the purpose of capturing and extracting intricate relationships among the identified main actors or objects.Consider the example of actors in a visual context, where the AAM can be exemplified through the perspective of an actors beholder, as depicted in Fig. 2.4. This illustration visually represents the functionality of the AAM, showcasing its ability to dynamically identify and focus on relevant actors or objects within the given context.

Initiating the process, the embedding of the environment feature f^e and actors features $\mathcal{F}^a$ into a unified dimensional space is facilitated through the application of multi-layer perceptrons (MLPs). These MLPs are intricately parameterized by θ_e and θ_a to ensure a seamless integration:

$$\hat{f}^e = MLP_{\theta_e}(f^e) \tag{2.2}$$

$$\hat{F}^a = \{\hat{f}_i^a\}_{i=1}^M \text{ where } \hat{f}_i^a = MLP_{\theta_a}(f_i^a) \tag{2.3}$$

Subsequently, the addition of $\hat{f}^e$ and each feature $\hat{f}_i^a$ from $\hat{F}^a$ is carried out through element-wise addition, denoted as $\oplus$, resulting in the creation of a collaborative feature. Following this, the attention score h_i^a associated with $\hat{f}_i^a$ can be determined by calculating the L2-norm of its respective collaborative feature. These computational procedures are succinctly expressed by the following equation:

$$h_i^a =|| \hat{f}_i^a \oplus \hat{f}^e ||_2 \tag{2.4}$$

As demonstrated in Malinowski et al., 2018, features possessing higher L2-norm values encapsulate more significant information and make a more substantial contribution to subsequent modules.

Following this, we normalize all L2-norm values using the softmax function to ensure a cumulative sum of 1.0, as L2-norm values are inherently unbounded:

$$H^a = \{h_i^a\}_{i=1}^M, \text{ where } h_i^a = \frac{e^{h_i^a}}{\Sigma_{i=1}^M e^{h_i^a}} \tag{2.5}$$

In order to extract the characteristics of any given number of primary actors, an adaptive threshold is established based on the total number of actors, denoted as $\tau = \frac{1}{|\mathcal{F}^a|}$. Subsequently, we selectively retrieve features $f_i^a \in \mathcal{F}^a$ only if their associated scores surpass τ:

$$\tilde{\mathcal{F}}^a = \{f_i^a \mid h_i^a \geq \tau\} \tag{2.6}$$

Following this step, we combine a collection of feature vectors for primary actors, denoted as $\tilde{\mathcal{F}}^a$, using the self-attention Transformer Encoder introduced in Vaswani et al., 2017a, resulting in a consolidated feature vector f^a.

For the objects beholder scenario, the input actors features $\mathcal{F}^a$ are substituted with the features of objects, denoted as $\mathcal{F}^o$.

2.4 Boundary-Matching Module (BMM)

The BMM module plays a crucial role in localizing action boundaries and proposing actions within videos. In our AOE-Net framework, we have incorporated the BMM module, drawing inspiration from established works such as BSN (T. Lin, X. Zhao, et al., 2018), BMN (T. Lin, X. Liu, et al., 2019), ABN (Vo-Ho et al., 2021), AEN (Vo, Yamazaki, et al., 2021), and AEI (Vo, Joo, et al., 2021) due to its standardized and straightforward design. The BMM module takes the output V-L features sequence $\mathcal{F} = \{f_i\}_{i=1}^T$ from the PMR module as input. Our BMM module consists of three integral components: semantic modeling, temporal estimation (TE), and proposal estimation (PE), as illustrated in Fig. 2.5. The initial component is dedicated to modeling the semantic

Table 2.1: The architecture of BMM comprises three components: $\mathcal{F}$, which represents the input feature derived from PMR; T, indicating the temporal length of the video; and D, representing the maximum duration of proposals in terms of the number of snippets.

Layers	**Input**	**Output**
1DConv. $256 \times 3/1$, ReLU	$\mathcal{F} : F \times T$	$O_1 : 256 \times T$
1DConv. $128 \times 3/1$, ReLU	$O_1 : 256 \times T$	$O_2 : 128 \times T$
1DConv. $256 \times 3/1$, ReLU	$O_2 : 128 \times T$	$O_3 : 256 \times T$
1DConv. $2 \times 3/1$, Sigmoid	$O_3 : 256 \times T$	$O_T : 2 \times T$
Matching layer	$O_2 : 128 \times T$	$O_5 : 128 \times 32 \times D \times T$
3DConv. $512 \times 32 \times 1 \times 1/(32,0,0)$, ReLU	$O_5 : 128 \times 32 \times D \times T$	$O_6 : 512 \times 1 \times D \times T$
squeeze	$O_6 : 512 \times 1 \times D \times T$	$O_7 : 512 \times D \times T$
2DConv. $128 \times 1 \times 1/(0,0)$, ReLU	$O_7 : 512 \times D \times T$	$O_8 : 128 \times D \times T$
2DConv. $128 \times 3 \times 3/(1,1)$, ReLU	$O_8 : 128 \times D \times T$	$O_9 : 128 \times D \times T$
2DConv. $2 \times 1 \times 1/(0,0)$, Sigmoid	$O_9 : 128 \times D \times T$	$O_P : 1 \times D \times T$

relationships among video snippets. The TE component evaluates each snippet $s_i \,|_{i=1}^{T}$ to determine the probabilities of action starting (P_i^S) and action ending (P_i^E) within s_i. Simultaneously, the PE component assesses every interval $[i,j]$ in the video to estimate its actionness score $P_{i,d}^A$,

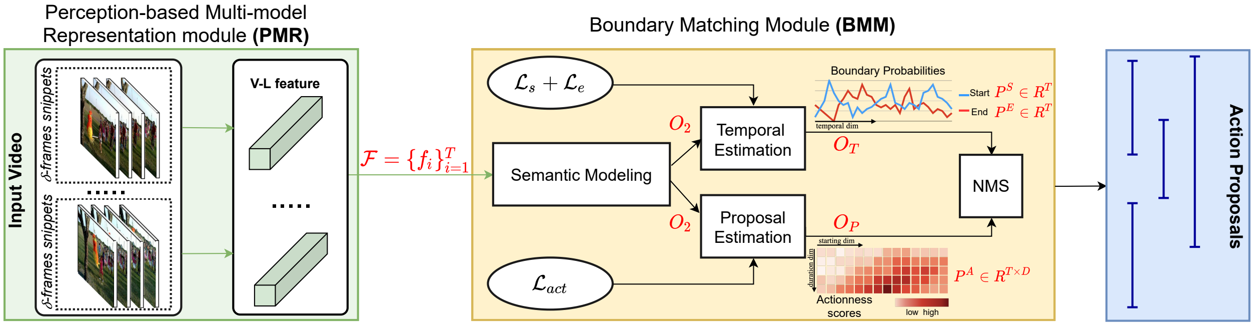


Figure 2.5: Our proposed AOE-Net encompasses two essential components: the perception-based multi-model representation module (PMR) and the boundary-matching module (BMM), contributing to its comprehensive architectural design.

where $d = j - i$. A detailed breakdown of the BMM architecture is presented in Table 2.1. The semantic modeling component is realized through two 1-D Conv. layers, producing a feature map $O_2 \in R^{128\times T}$. Subsequent components, TE and PE, take O_2 as input, generating $O_T \in R^{2\times T}$ and $O_P \in R^{1\times D\times T}$, respectively. The output O_T signifies the probabilities of action starts ($P^S \in R^T$) and action ends ($P^E \in R^T$), while the output O_P encompasses actionness scores $P^A \in R^{D\times T}$.

During the inference stage, a search is conducted through P^S and P^E to identify temporal locations i with local maxima, forming sets of potential starting and ending temporal locations, denoted as P_i^S and P_i^E respectively. Subsequently, pairing starting and ending locations (s, e), where $s \leq e \leq T$, results in the creation of candidate proposals. Each candidate proposal is assigned a score $s = P_s^S \cdot P_e^E \cdot P_{s,e-s}^A$. Finally, leveraging the timestamps and scores of these candidate proposals, Non-Maximum Suppression (NMS) (Bodla et al., 2017; Neubeck and Van Gool, 2006) is applied to generate the ultimate set of temporal action proposals.

2.5 Experiments

Datasets and Metrics

Datasets

Our experimentation involves TAPG and TAD , utilizing both ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015) and THUMOS-14 Jiang et al., 2014 datasets . ActivityNet-1.3 comprises 20,000 videos and 200 annotated activities, while THUMOS-14 includes 414 videos featuring 20 distinct types of actions. Video preprocessing, with a snippet length of $\delta = 16$, aligns with the methodologies of previous studies T. Lin, X. Zhao, et al., 2018; T. Lin, X. Liu, et al., 2019; C. Lin, J. Li, et al., 2020 across all experiments. To showcase the efficacy of our proposed AOE-Net on ego-centric videos, we further extend our investigation to the TAPG task within the EPIC-KITCHENS 100 dataset (Damen, Doughty, et al., 2021) . This dataset encompasses 100 hours of video, spanning 20 million frames and 90,000 actions within 700 variable-length videos. The recordings were captured in 45 diverse environments using head-mounted cameras.

Metrics

Within the framework of TAPG, the assessment of the proposed AOE-Net and its comparison with state-of-the-art (SOTA) approaches relies on two commonly employed metrics: AR@AN and AUC. The former, denoted as Average Recall (AR), is computed at a specific average number of proposals (AN) preserved within each video. Meanwhile, the latter, Area Under the Curve (AUC), represents the score derived from the curve plotting AR against AN. Notably, AR@100 and AUC stand out as the predominant metrics within the context of ActivityNet-1.3. In the case of THUMOS-14, the evaluation solely incorporates AR@AN for method comparison, with multiple AN selected from a predefined list comprising [50, 100, 200, 500, 1000].

In the realm of TAD, the benchmarking of approaches centers around the mean Average Precision (mAP). Following established conventions, as outlined in previous works such as T. Lin, X. Zhao, et al., 2018; C. Lin, J. Li, et al., 2020; T. Lin, X. Liu, et al., 2019; S. Liu, X. Zhao, et al., 2020; P. Zhao, Xie, et al., 2020a, TAD methods undergo evaluation in the ActivityNet-1.3 dataset with temporal Intersection over Union (tIoU) thresholds of $\{0.5, 0.75, 0.95\}$, culminating in an averaged mAP. In contrast, TAD methods evaluated within the THUMOS-14 dataset are subjected to tIoU thresholds of $\{0.3, 0.4, 0.5, 0.6, 0.7\}$.

Implementation Details

For the extraction of visual features from videos, the foundational network employed in all experiments on ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015) and THUMOS-14 (Jiang

Methods	Venue & Year	Feature	AR@100	AUC(val)	AUC(test)
TCN (Dai, Singh, et al., 2017)	ICCV17	2Stream	–	59.58	61.56
MSRA (Yao, Y. Li, Qiu, et al., 2017)	CVPRW17	P3D	–	63.12	64.18
SSTAD (Buch et al., 2017)	BMVC17	C3D	73.01	64.40	64.80
CTAP (Jiyang Gao, K. Chen, and Nevatia, 2018)	ECCV18	2Stream	73.17	65.72	–
BSN (T. Lin, X. Zhao, et al., 2018)	ECCV18	2Stream	74.16	66.17	66.26
SRG (Eun et al., 2019)	IEEE-TCSVT19	2Stream	74.65	66.06	–
MGG (Y. Liu, Ma, et al., 2019)	CVPR19	I3D	74.54	66.43	66.47
BMN (T. Lin, X. Liu, et al., 2019)	ICCV19	2Stream	75.01	67.10	67.19
DBG (C. Lin, J. Li, et al., 2020)	AAAI20	2Stream	76.65	68.23	68.57
BSN++ (Su, Gan, et al., 2021)	ACCV20	2Stream	76.52	68.26	–
TSI++ (S. Liu, X. Zhao, et al., 2020)	ACCV20	2Stream	76.31	68.35	68.85
MR (P. Zhao, Xie, et al., 2020a)	ECCV20	I3D	75.27	66.51	–
AEN (Vo-Ho et al., 2021)	ICASSP21	C3D	75.65	68.15	68.99
ABN (Vo, Yamazaki, et al., 2021)	IEEE-Access21	C3D	76.72	69.16	69.26
SSTAP (X. Wang et al., 2021)	CVPR21	I3D	75.54	67.53	–
TCANet (Qing et al., 2021)	CVPR21	2Stream	76.08	68.08	–
Zheng, et.al. (Zheng, D. Chen, and Hu, 2021)	NPL21	2Stream	74.93	65.20	–
AEI (Vo, Joo, et al., 2021)	BMVC21	C3D	*77.24*	*69.47*	*70.09*
AOE-Net		C3D	**77.67**	**69.71**	**70.10**
		2Stream	76.32	68.35	69.00
		SlowFast	76.95	68.95	69.86

Table 2.2: **TAPG** performance comparisons were conducted on ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015) with a focus on AR@100 and AUC metrics. The evaluations were carried out on both the validation set, considering AR@100 and AUC, and the testing set, focusing on AUC.

et al., 2014) is a C3D network (Ji et al., 2013) pretrained on the Kinetics-400 dataset (Kay et al., 2017). The dimensions of the features derived from the C3D backbone amount to 2048.

Within the realm of object perception, the extraction of object text relies on CLIP (Radford et al., 2021), pretrained on a substantial dataset comprising 400 million image-text pairs sourced from the Internet. Encoding of the text feature and image feature is carried out by Transformer (Vaswani et al., 2017a) and Vision Transformer (Dosovitskiy et al., 2021) networks, respectively. For the identification of humans in the actors' domain, a Faster-RCNN model (Ren et al., 2015), pretrained on the COCO dataset (T.-Y. Lin et al., 2014), is employed. The AOE-Net is trained using the Adam optimizer, with an initial learning rate of 0.0001 for ActivityNet-1.3 and 0.001 for THUMOS-14.

In the context of ActivityNet-1.3, Soft-NMS (SNMS) (Bodla et al., 2017) is applied during post-processing across all TAPG and TAD experiments. On THUMOS-14, in accordance with T. Lin, X. Zhao, et al., 2018; T. Lin, X. Liu, et al., 2019, both Soft-NMS (Bodla et al., 2017) and NMS

(Neubeck and Van Gool, 2006) are employed in the post-processing of TAPG, whereas TAD utilizes only NMS. The subsequent presentation of experimental results highlights the best performance in **bold** and the second-best performance in *underline*.

Performance and Comparison on TAPG

Table 2.2 showcases the TAPG comparison results on the validation and testing sets of ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015). Our AOE-Net, utilizing C3D (Ji et al., 2013) features, exhibits superior performance over existing methods, demonstrating a notable margin in terms of AR@100 and AUC. The comparative analysis in Table 2.3 extends to THUMOS-14, where AOE-Net competes favorably with other TAPG methods. Specifically, on SNMS, AOE-Net secures the second-best position across all AR@ANs, except for AR@100, where it competes closely with the top performer (50.26 vs. 50.67). On NMS, AOE-Net achieves the best performance on AR@100 and ranks second on AR@200 and AR@500, with minimal gaps compared to the state-of-the-art (57.49 vs. 57.74 and 62.40 vs. 62.74, respectively). Notably, the TAPG performance of our AOE-Net on both datasets stands out competitively, closely trailing AEI-B (Vo, Joo, et al., 2021) and ABN (Vo, Yamazaki, et al., 2021), which also incorporate local actors and the global environment. This experimentation strongly validates our choice of leveraging human perception principles for analyzing human actions in untrimmed videos.

In addition to the exclusive assessment of AOE-Net on TAPG and TAD tasks, it is imperative to explore the impact of various backbone features on our AOE-Net. The performance of our proposed AOE-Net network is analyzed concerning different features, namely C3D (Ji et al., 2013), 2Stream (Simonyan and Zisserman, 2014), and Slowfast (Feichtenhofer, Fan, et al., 2019), each possessing feature dimensions of 2048, 2314, and 400, respectively. The outcomes are presented in the lower section of Table 2.2 for the TAPG task within the ActivityNet-1.3 dataset (Fabian Caba Heilbron and Niebles, 2015). Upon examination, it is evident that the performance using C3D (Ji et al., 2013) features attains a state-of-the-art status, while the performance with SlowFast (Feichtenhofer, Fan, et al., 2019) features closely trails behind. Conversely, the utilization of 2Stream (Simonyan and Zisserman, 2014) features yields the least favorable performance among the three types of

	Methods	Venue & Year	Feature	@50	@100	@200	@500	@1000	Average
SNMS	CTAP (Jiyang Gao, K. Chen, and Nevatia, 2018)	ECCV18	2Stream	32.49	42.61	51.97	–	–	–
	BSN (T. Lin, X. Zhao, et al., 2018)	ECCV18	2Stream	37.46	46.06	53.21	60.64	64.52	52.38
	MGG (Y. Liu, Ma, et al., 2019)	CVPR19	I3D	39.93	47.75	54.65	61.36	64.06	53.55
	BMN (T. Lin, X. Liu, et al., 2019)	ICCV19	2Stream	39.36	47.72	54.70	62.07	65.49	53.87
	DBG (C. Lin, J. Li, et al., 2020)	AAAI20	2Stream	37.32	46.67	54.50	62.21	66.40	53.42
	Rapnet (Jialin Gao et al., 2020)	AAAI20	C3D	40.35	48.23	54.92	61.41	–	–
	TSI++(S. Liu, X. Zhao, et al., 2020)	ACCV20	2Stream	42.30	*50.51*	57.24	63.43	–	–
	MR(P. Zhao, Xie, et al., 2020a)	ECCV20	I3D	44.23	**50.67**	55.74	–	–	–
	BC-GNN (Bai et al., 2020)	ECCV20	2Stream	40.50	49.60	56.33	62.80	–	–
	TCANet (Qing et al., 2021)	CVPR21	2Stream	42.05	50.48	57.13	63.61	66.88	*56.03*
	SSTAP (X. Wang et al., 2021)	CVPR21	2Stream	41.01	50.12	56.69	–	**68.81**	–
	ABN (Vo, Yamazaki, et al., 2021)	IEEE-Access21	C3D	40.87	49.09	56.24	63.53	67.29	55.40
	AEI-B (Vo, Joo, et al., 2021)	BMVC21	C3D	**44.97**	50.13	**57.34**	**64.43**	67.78	**56.93**
	AOE		C3D	*44.56*	50.26	*57.30*	*64.32*	*68.19*	**56.93**
NMS	BSN (T. Lin, X. Zhao, et al., 2018)	ECCV18	C3D	27.19	35.38	43.61	53.77	59.50	43.89
	BSN (T. Lin, X. Zhao, et al., 2018)	ECCV18	2Stream	35.41	43.55	52.23	61.35	*65.10*	51.53
	BMN (T. Lin, X. Liu, et al., 2019)	ICCV19	C3D	29.04	37.72	46.79	56.07	60.96	46.12
	BMN (T. Lin, X. Liu, et al., 2019)	ICCV19	2Stream	37.15	46.75	54.84	62.19	**65.22**	53.23
	DBG (C. Lin, J. Li, et al., 2020)	AAAI20	C3D	32.55	41.07	48.83	57.58	59.55	47.92
	DBG (C. Lin, J. Li, et al., 2020)	AAAI20	2Stream	40.89	49.24	55.76	61.43	61.95	53.85
	ABN (Vo, Yamazaki, et al., 2021)	IEEE-Access21	C3D	*44.89*	51.86	57.36	61.67	62.59	55.67
	AEI-B (Vo, Joo, et al., 2021)	BMVC21	C3D	**45.74**	*52.39*	**57.74**	**62.49**	63.38	**56.35**
	AOE		C3D	44.78	**52.41**	*57.49*	*62.40*	63.40	*56.10*

Table 2.3: **TAPG** performance evaluation on **THUMOS-14** based on AR@AN, with SNMS denoting Soft-NMS (Bodla et al., 2017).

Methods	Venue & Year	Feature	0.50	0.75	0.95	Average
BSN (T. Lin, X. Zhao, et al., 2018)	ECCV18	2Stream	46.45	29.96	8.02	30.03
GTAN (Long et al., 2019)	CVPR19	P3D	*52.61*	34.14	8.91	34.31
BMN (T. Lin, X. Liu, et al., 2019)	ICCV19	2Stream	50.07	34.60	8.29	33.85
GTAD (Xu, C. Zhao, et al., 2020)	CVPR20	2Stream	50.36	34.60	9.02	34.09
P-GCN (Zeng et al., 2019)	CVPR20	I3D	42.90	28.14	2.47	26.99
MR (P. Zhao, Xie, et al., 2020a)	ECCV20	2Stream	43.47	33.91	9.21	30.12
TSI++ (S. Liu, X. Zhao, et al., 2020)	ACCV20	2Stream	51.18	*35.00*	6.59	34.15
BC-GNN (Bai et al., 2020)	ECCV20	2Stream	50.56	34.75	9.37	34.26
RTD (Tan et al., 2021)	ICCV21	2Stream	47.21	30.68	8.61	30.83
ABN (Vo, Yamazaki, et al., 2021)	IEEE-Access21	C3D	51.78	34.18	*10.29*	34.22
AEI-B (Vo, Joo, et al., 2021)	BMVC21	C3D	52.3	34.5	9.7	**34.7**
AOE		C3D	**54.42**	**35.43**	**10.35**	*34.48*

Table 2.4: **Comparison of TAD Results on ActivityNet-1.3:** The evaluation focuses on mAP@tIoU and mAP metrics, with the integration of proposals alongside video-level classification outcomes derived from Xiong et al., 2016.

backbone features.

In the context of TAPG, *generalizability* emerges as a crucial criterion for assessing the efficacy of a method. Adhering to the experimental framework outlined in T. Lin, X. Zhao, et al., 2018; T. Lin, X. Liu, et al., 2019; C. Lin, J. Li, et al., 2020; S. Liu, X. Zhao, et al., 2020; Vo, Yamazaki, et al., 2021, our investigation unfolds within the domain of ActivityNet-1.3, encompassing two distinct subsets: *Seen* comprising "Sports, Exercises, and Recreation," and *Unseen* encompassing "Socializing, Relaxing, and Leisure." The training of our AOE-Net is executed on both *Unseen+Seen* and *Seen* training sets independently. Subsequently, evaluations are conducted on the *Seen* and *Unseen* validation sets. The performance comparison and visualization between AOE-Net and other state-of-the-art (SOTA) methods are illustrated in Fig. 2.6. In each chart, the last columns depict the performance of AOE-Net, showcasing its superiority over other SOTA methods. Notably, Fig. 2.6 underscores that AOE-Net consistently achieves commendable performances on the *Seen* validation set, with an acceptable decline on the *Unseen* validation set in both training configurations. This observation suggests the high generalizability of our AOE-Net to previously unseen action types.

Performance and Comparison on TAD

To ensure a fair comparison, we adhere to the experimental configurations outlined in prior works T. Lin, X. Zhao, et al., 2018; T. Lin, X. Liu, et al., 2019; C. Lin, J. Li, et al., 2020; Xu, C. Zhao,

Methods		Evaluation			
		Seen		Unseen	
	Training	AR@100	AUC	AR@100	AUC
BSN (T. Lin, X. Zhao, et al., 2018)	Seen + Unseen	72.40	63.80	71.84	63.99
	Seen	72.42	64.02	71.32	63.38
BMN (T. Lin, X. Liu, et al., 2019)	Seen + Unseen	72.96	65.02	72.68	65.05
	Seen	72.47	64.37	72.46	64.47
TSI++ (S. Liu, X. Zhao, et al., 2020)	Seen + Unseen	74.69	66.54	74.31	66.14
	Seen	73.59	65.60	73.07	65.05
DBG (C. Lin, J. Li, et al., 2020)	Seen + Unseen	73.30	66.57	67.23	64.59
	Seen	72.95	66.23	64.77	62.18
ABN (Vo, Yamazaki, et al., 2021)	Seen + Unseen	74.58	66.96	75.25	67.49
	Seen	74.40	66.69	73.66	65.49
AOE-Net	Seen + Unseen	76.36	68.31	77.31	69.07
	Seen	76.43	68.42	74.90	66.92

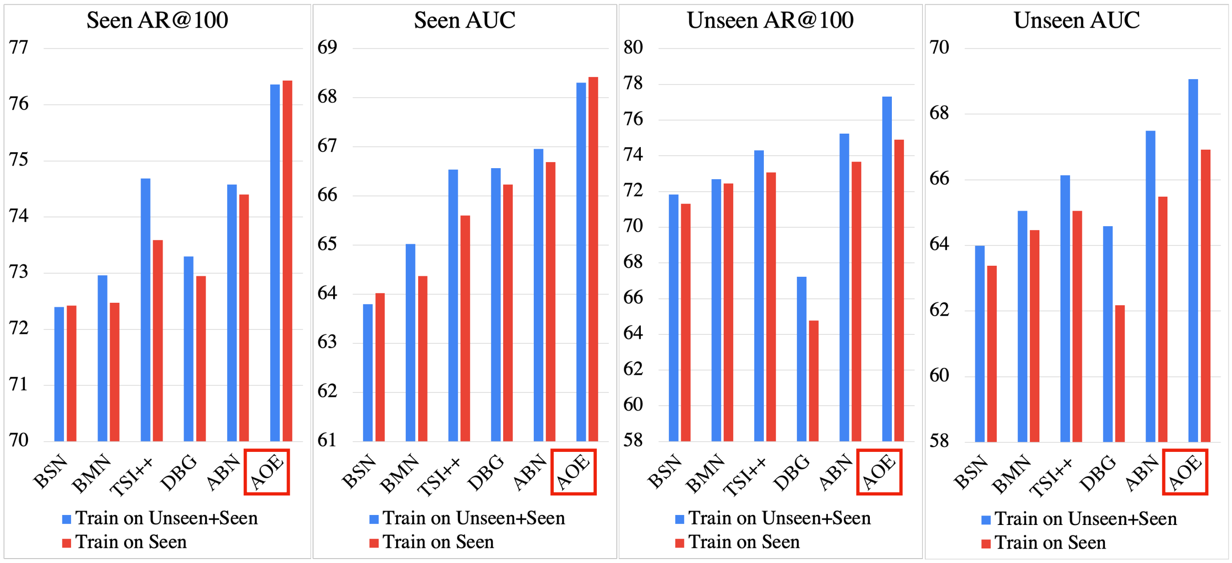


Figure 2.6: **Generalizability** assessment and comparisons were conducted on ActivityNet-1.3, focusing on AR@100 and AUC metrics. The training methods were implemented on two distinct sets: *Unseen+Seen* and *Seen*. The subsequent evaluation was carried out on both *Seen* (depicted in the first two charts) and *Unseen* (depicted in the last two charts) validation sets. The top section presents a detailed breakdown of the individual performance across various experimental settings for each method. The bottom section features a visual representation of the generalizability comparison between our proposed AOE-Net and other methods.

et al., 2020; Bai et al., 2020; Y. Liu, Ma, et al., 2019; Tan et al., 2021; Vo, Yamazaki, et al., 2021 when annotating action proposals generated by our AOE-Net. In the case of ActivityNet-1.3, we utilize the top-1 video-level classification outcomes produced by the approach detailed in Xiong et al., 2016 to label our proposals. Conversely, for THUMOS-14, we assign labels to our action

	Methods	Year	Feature	0.7	0.6	0.5	0.4	0.3	Average
UntrimmedNet	BSN (T. Lin, X. Zhao, et al., 2018)	ECCV18	2Stream	20.0	28.4	36.9	45.0	53.5	36.76
	BMN (T. Lin, X. Liu, et al., 2019)	ICCV19	2Stream	20.5	29.7	38.8	47.4	56.0	38.48
	MGG (Y. Liu, Ma, et al., 2019)	CVPR19	2Stream	21.3	29.5	37.4	46.8	53.9	37.78
	GTAN (Long et al., 2019)	CVPR19	P3D	–	–	38.8	47.2	57.8	–
	DBG (C. Lin, J. Li, et al., 2020)	AAAI20	2Stream	21.7	30.2	39.8	49.4	57.8	39.78
	GTAD (Xu, C. Zhao, et al., 2020)	CVPR20	2Stream	23.4	30.8	40.2	47.6	54.5	39.30
	TSI++ (S. Liu, X. Zhao, et al., 2020)	ACCV20	2Stream	22.4	33.2	42.6	52.1	*61.0*	42.26
	BC-GNN (Bai et al., 2020)	ECCV20	2Stream	23.1	31.2	40.4	49.1	57.1	40.18
	BU-TAL (P. Zhao, Xie, et al., 2020b)	ECCV20	2Stream	**28.5**	*38.0*	45.4	50.7	53.9	43.30
	TCANet (Qing et al., 2021)	CVPR21	2Stream	26.7	36.8	44.6	53.2	60.6	44.38
	RTD (Tan et al., 2021)	ICCV21	2Stream	25.0	36.4	45.1	53.1	58.5	43.62
	ABN (Vo, Yamazaki, et al., 2021)	IEEE-Access21	C3D	25.6	37.0	*46.1*	*54.0*	59.9	44.51
	AEI-B (Vo, Joo, et al., 2021)	BMVC21	C3D	23.4	35.9	44.7	52.7	58.7	43.08
	AOE-Net	–	C3D	*25.8*	**38.8**	**48.4**	**57.3**	**63.4**	**46.74**
P-GCN	BSN (T. Lin, X. Zhao, et al., 2018)	ECCV18	I3D	–	–	49.1	57.8	63.6	–
	MR (P. Zhao, Xie, et al., 2020a)	ECCV20	2Stream	–	–	50.10	60.99	66.29	–
	GTAD (Xu, C. Zhao, et al., 2020)	CVPR20	2Stream	22.9	37.6	**51.6**	*60.4*	*66.4*	*47.78*
	AOE-Net	–	C3D	**23.5**	37.4	*50.9*	**60.6**	**67.1**	**47.89**

Table 2.5: **TAD** evaluations were conducted on the **THUMOS-14** dataset, assessing mAP@tIoU with two distinct classifiers: UntrimmedNet (L. Wang et al., 2017) and P-GCN (Zeng et al., 2019).

proposals based on either UntrimmedNet (L. Wang et al., 2017) (leveraging top-2 classification results) or P-GCN (Zeng et al., 2019).

Table 2.4 illustrates a comprehensive performance evaluation of TAD between AOE-Net and several state-of-the-art (SOTA) methods on the ActivityNet-1.3 validation set. The outcomes highlight the superiority of our method across various temporal Intersection over Union (tIoU) thresholds when compared to other SOTA techniques. The experiment outcomes presented in Table 2.5 for the THUMOS-14 test set further affirm the effectiveness of our AOE-Net, showcasing its superiority over other SOTA methods across a majority of metrics when employing both classifiers.

2.6 Ablation Study

An extensive ablation study is undertaken to demonstrate the efficacy of individual components within the proposed AOE-Net, along with assessing the resilience of AOE-Net in the context of egocentric videos. Furthermore, we present findings on the network efficiency and AOE-Net performance across various configurations of the hyper-parameter K. Further details of the ablation study will be provided in the supplementary materials.

Exp	Setting					TAPG Performance				
	Act.	Env.	Obj.	AAM	Soft-Att	@50	@100	@200	@500	@1000
#1	√	×	×	×	√	25.96	35.14	43.48	52.37	58.47
#2	×	√	×	×	×	38.94	47.80	54.93	61.92	65.96
#3	×	×	√	×	√	18.06	26.68	37.14	49.28	56.99
#4	√	√	×	×	√	40.87	49.09	56.24	63.53	67.29
#5	√	√	√	×	√	42.60	49.86	56.87	63.76	67.60
#6	√	√	×	√	×	43.79	49.67	56.73	63.49	67.36
#7	√	√	√	√	×	44.56	50.26	57.30	64.32	68.19

Table 2.6: The comparisons of TAPG across various network settings involve the examination of actors (Act.), environment (Env.), and objects (Obj.) as beholders.

Attention	THUMOS-14					ActivityNet-1.3		
	@50	@100	@200	@500	@1000	AR @100	AUC (val)	AUC (test)
Hard (Malinowski et al., 2018)	43.74	49.24	56.63	63.46	67.25	77.11	69.02	69.56
Soft (Vaswani et al., 2017a)	42.60	49.86	56.87	63.76	67.60	76.93	69.06	69.23
AAM	**44.56**	**50.26**	**57.30**	**64.32**	**68.19**	**77.67**	**69.71**	**70.10**

Table 2.7: TAPG involves a comparison between AAM with attention, as discussed in Malinowski et al., 2018 and Vaswani et al., 2017a.

	AR@10	AR@100	AUC
BMN (T. Lin, X. Liu, et al., 2019)	11.59	34.26	25.14
AOE-Net	**15.99**	**37.40**	**29.20**

Table 2.8: In the realm of Temporal Action Proposal Generation (TAPG), we assess our AOE-Net alongside BMN (T. Lin, X. Liu, et al., 2019) in the context of egocentric videos(Damen, Doughty, et al., 2021).

Contribution of each beholder

We analyze the performance of TAPG on THUMOS-14 using various network settings, as outlined in Table 2.6. Experiments (#1-3) showcase the individual contributions of each observer, while experiments (#4-7) present different combinations of features, underscoring the significant role of actors and objects in comprehending human action. Comparisons between experiments (#4 vs. #6) and (#5 vs. #7) underscore the substantial impact of AAM.

In Exp.#1 and Exp.#3, the absence of the Environment Beholder precludes the application of AAM, which necessitates environment features. To address this, AAM is substituted with a straightforward soft self-attention layer, followed by an average pooling operation to amalgamate multiple actors. Similarly, in Exp.#4 and Exp.#5, the aforementioned substitution strategy is employed to accentuate the effectiveness of AAM.

Effectiveness of AAM

We proceed with our investigation into the efficacy of the proposed AAM in the TAPG task, evaluating its performance on both ActivityNet-1.3 and THUMOS-14. This assessment involves comparing AAM with alternative attention mechanisms, namely soft self-attention (Vaswani et al., 2017a) (Soft) and hard attention (Malinowski et al., 2018) (Hard), as depicted in Table.2.7.

In the case of the soft self-attention mechanism, we streamline our AAM by omitting the initial hard attention component defined in Eq. 2.2-2.6. Instead, we directly input the set of actor features $\mathcal{F}^a$ (or object features $\mathcal{F}^o$) into a self-attention mechanism.

Conversely, for the hard-attention mechanism, we replace the self-attention segment at the conclusion of our AAM with a straightforward average pooling operation. This operation averages the selected actor features $\tilde{F}^a$ (or selected object features $\tilde{F}^o$) into a singular representation f^a (or f^o).

The superior performances evident on both datasets, as illustrated in Table 2.7, affirm the compelling advantages of AAM over soft self-attention and hard attention mechanisms.

Performance of AOE-Net with different number of objects

The hyper-parameter K, denoting the number of input objects for the objects beholder (see Subsection 2.2), plays a crucial role in influencing the performance of our AOE-Net. A larger value of K introduces more noisy information into the overall model, stemming from inaccurately detected objects. Conversely, a smaller value of K results in an insufficient presentation of significant information by the objects beholder, hindering its contribution to action comprehension.

In the conducted ablation study, we evaluate the performance of AOE-Net across various values of

Number of Objects (K)	AR@100	AUC (val)	AUC (test)
0	77.02	68.98	69.72
1	77.15	69.17	69.95
5	77.45	69.43	69.96
10	77.24	69.26	69.56
20	**77.67**	*69.71*	*70.10*
30	77.55	69.63	69.96
40	**77.67**	**69.86**	**70.22**
50	77.24	69.17	69.81

Table 2.9: Assessing the TAPG performance of our AOE-Net on ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015) using different K settings.

K in the context of the TAPG task, utilizing the ActivityNet-1.3 dataset Fabian Caba Heilbron and Niebles, 2015. The comparative results are presented in Table 2.9.

The findings in Table 2.9 demonstrate a positive correlation between the increment of *K* and the improvement in the TAPG performance of AOE-Net. Nevertheless, beyond $K > 20$, the performance exhibits fluctuations and lacks robustness, attributed to an increased presence of wrongly detected objects in each snippet. Consequently, we deduce that maintaining $K = 20$ strikes the optimal balance between performance and robustness for our AOE-Net.

Robustness of AOE-Net to egocentric videos

To assess AOE-Net's robustness in handling egocentric videos, we employ EPIC-KITCHENS 100 (Damen, Doughty, et al., 2021) as the benchmark for the TAPG task. The TAPG performance is presented in Table 2.8, showcasing a comparison between our AOE-Net and BMN (T. Lin, X. Liu, et al., 2019). Despite the absence of visible actors in the egocentric videos, AOE-Net demonstrates commendable TAPG performance, exhibiting a significant improvement over BMN. This underscores the effectiveness of the objects' perspective in our approach.

Network efficiency

The efficiency comparison between AOE-Net and the previous state-of-the-art (SOTA) models, along with the number of parameters (in millions, M), computational cost (GFLOPs), and inference

	#Params	FLOPs	Inference time (s)	
	(M)	(G)	GPU	CPU
BMN (T. Lin, X. Liu, et al., 2019)	4.9	71.22	0.128	4.15
DBG (C. Lin, J. Li, et al., 2020)	2.9	47.52	0.03	-
GTAD (Xu, C. Zhao, et al., 2020)	5.6	150.28	0.14	-
ABN (Vo, Yamazaki, et al., 2021)	6.9	87.88	0.07	0.21
AEI (Vo, Joo, et al., 2021)	6.9	90.62	0.08	0.21
AOE-Net	8.8	94.02	0.12	0.27

Table 2.10: The network efficiencies of AOE-Net and various prior studies are compared in this analysis.

time on a 3-minute video, is presented in Table 2.10. The evaluation was conducted on both an Intel Core i9-9920X CPU and a single NVIDIA RTX 2080 Ti.

Qualitative Analysis of AAM

Qualitative Results of AAM with Actors Beholder:

In Fig. 2.7, we present a visual representation showcasing the qualitative performance of our proposed Actor Attention Module (AAM) in the context of identifying main actors within a set of detected individuals. The video samples used for this evaluation are sourced from ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015).

When faced with scenarios involving the detection of multiple actors, as depicted in Fig. 2.7(a) and Fig. 2.7(c), our AAM effectively excels in the task of discerning and selecting the primary actors while filtering out insignificant individuals. This functionality aims to eliminate redundant information, ensuring that only the most pertinent actors are considered for input into the subsequent boundary-matching module.

Fig. 2.7(b) illustrates a situation where the environmental context may be mundane and contribute minimally to action perception. Nevertheless, the local information within the bounding box surrounding the main actor proves valuable in emphasizing the action. In this case, AAM once again demonstrates its effectiveness in selecting the main actor engaged in the action, thus highlighting its capability to discern and prioritize relevant information.

Performance of AAM affected by human detector:

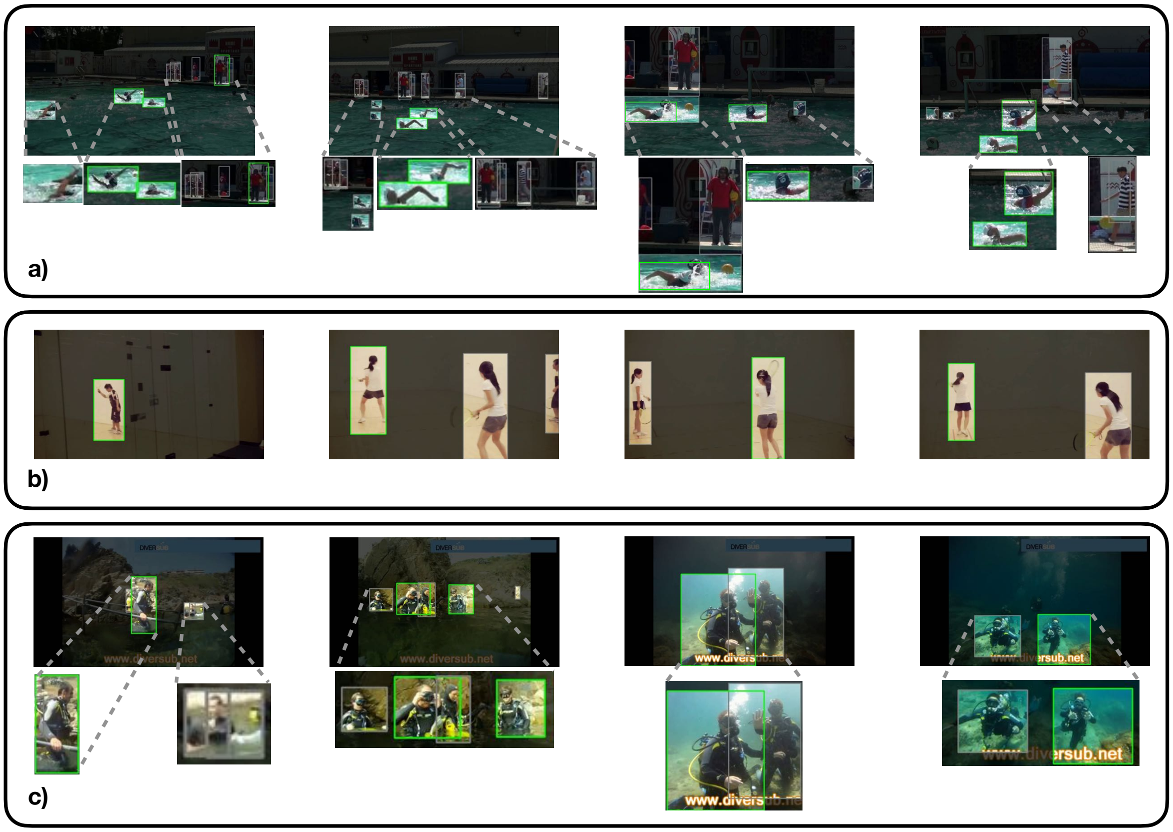


Figure 2.7: The depiction illustrates the main actors selected through the application of AAM on ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015). Three distinct videos, denoted as (a), (b), and (c), are showcased. The backdrop is rendered in black, with the bounding boxes of primary actors demarcated by a green line, while the bounding boxes of less significant actors are delineated by a grey line.

The Faster-RCNN (Ren et al., 2015) model, trained on the COCO dataset (T.-Y. Lin et al., 2014), serves as the human detector in our study. However, its performance is not flawless in video scenarios, often affected by issues such as motion blurs or low resolutions. Consequently, the quality of the detected human bounding boxes directly impacts the performance of the Action Attention Module (AAM).

In Fig. 2.8, we present frames from four videos where the human detector exhibits suboptimal performance in producing accurate human bounding boxes. In Fig. 2.8(a), the green bounding box encompasses two athletes in a pool, each enclosed in a separate bounding box. Despite the incorrect nature of the green bounding box, containing multiple humans (even three, if considering the one behind), it intuitively captures richer scene information than the individual boxes for each athlete. This illustrates the AAM's effectiveness in learning to select the most informative bounding boxes, irrespective of their quality in terms of human detection.

Fig. 2.8(b), (c), and (d) showcase instances where the human detector poorly localizes bounding boxes, capturing only body parts instead of the entire human. Remarkably, the AAM demonstrates its ability to learn and filter out these inaccuracies, selecting only the correctly localized bounding boxes.

From these observations, it becomes evident that the AAM can learn to avoid selecting poorly localized bounding boxes that do not fully encompass the humans. Interestingly, it also learns to select some inaccurately detected bounding boxes, which nonetheless contain more meaningful information than the correctly localized ones.

In conclusion, our study reveals that relying solely on the human detector to provide locations for attention hinders the AAM from reaching its maximum potential. Thus, we advocate for the development of a more sophisticated module in future research, one that can effectively localize interesting spatial locations in video frames, surpassing the limitations of the human detector.

Qualitative Analysis of Objects Beholder

Figure 2.9 illustrates the utilization of Objects Beholder by AOE-Net. The figure presents two videos representing two distinct categories: (A) visible actors and (B) non-visible actors. In (A), the actors are visible and engaged in the action of tightrope walking. Consequently, AOE-Net can leverage all the beholders available. On the other hand, in (B), the actors are not visible in the video frame but are involved in the action of cooking. In this scenario, AOE-Net can only depend on Objects Beholder.

In the depiction of both (A) and (B), we observe scenarios where the action is absent (A.i and B.i) and instances when the action is occurring (A.ii and B.ii).

In Fig. 2.9.(A), a noticeable distinction is evident between the objects identified in non-action

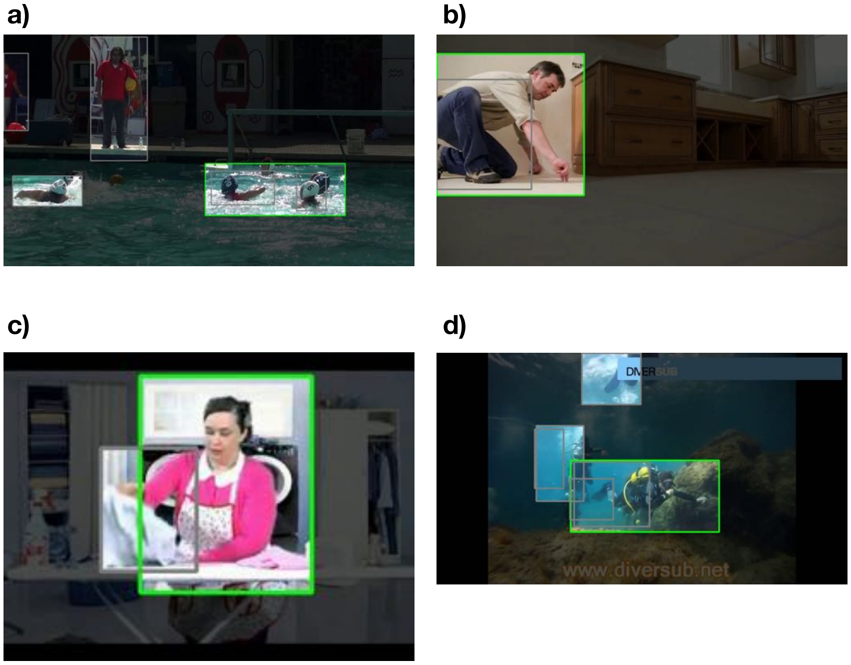


Figure 2.8: Illustration of AAM applied to ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015), focusing on instances where the human detector inadequately produces human bounding boxes. The backdrop is darkened, with the primary actors' bounding boxes delineated by a green line, while the bounding boxes of less significant actors are demarcated by a grey line.

and action scenarios. In particular, as depicted in Fig. 2.9(A.i), the non-action scenario features objects such as "City," "Building," and "Tower." Conversely, in Fig. 2.9(A.ii), the action scenario encompasses objects like "Rooftop," "Tightrope," "Roof," and "Hanging."

Similarly, as depicted in Fig. 2.9.(B), there is a notable distinction in the detected objects between non-action and action scenarios. In Fig. 2.9(B.i), the identified objects include "Kitchen," "Cooker," "Oven," and "Stove," among others. Conversely, in Fig. 2.9(B.ii), the objects in the action scene encompass "Passata," "Chorizo," "Pan," and "Salsa," among others.

Qualitative Analysis of AOE-Net

Figure 2.10 presents the qualitative outcomes of our AOE-Net in TAPG of ActivityNet-1.3 (Fabian Caba Heilbron and Niebles, 2015), juxtaposed with the performance of prior state-of-the-art approaches T. Lin, X. Liu, et al., 2019; C. Lin, J. Li, et al., 2020; Vo, Yamazaki, et al., 2021. The illustration includes two instances of medium difficulty, a challenging case, and a scenario involving egocentric video. For each video, we have chosen to showcase proposals from all methods that

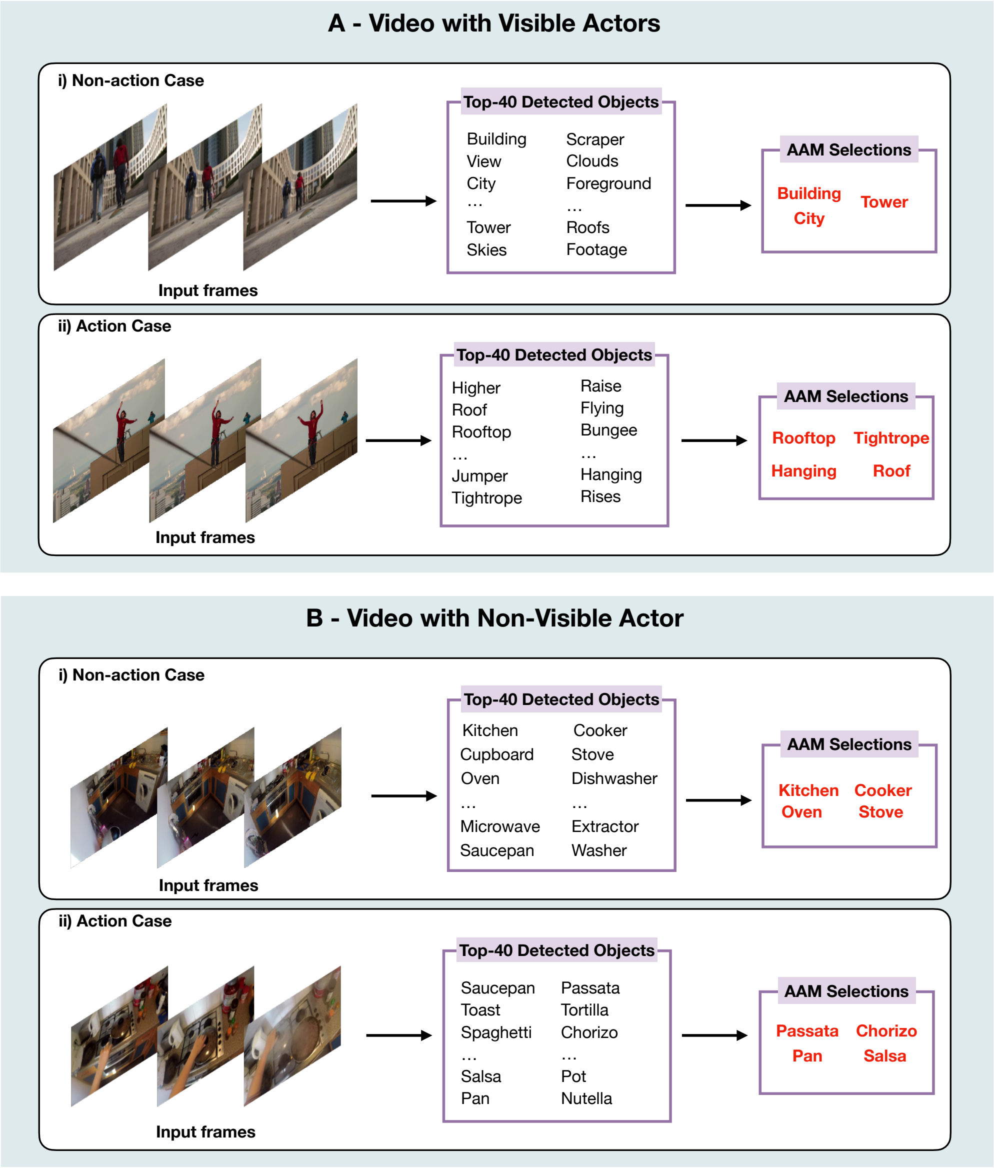


Figure 2.9: Demonstrating the efficacy of Objects Beholder with AAM is achieved through qualitative results in (A) videos featuring visible actors and (B) videos with non-visible actors. For both scenarios, we present two instances—one with action and one without action. The left side displays the input frames, the middle exhibits the objects detected by CLIP, and the right showcases the most pertinent objects selected by AAM.

possess scores exceeding 0.4 in the qualitative examples.

In the examination of uncomplicated scenarios through video analysis, as depicted in Fig. 2.10-A, it is readily discernible from the entire frame which actors are undergoing piercing (or trimming in video (b)). However, pinpointing the precise moment of the piercing action poses a challenge due to the involvement of the doctor’s hand (or the tailor in video (b)), positioned beyond the video frame. Consequently, existing models such as BMN (T. Lin, X. Liu, et al., 2019) and DBG (C. Lin, J. Li, et al., 2020) falter in accurately proposing action intervals. Similarly, ABN (Vo, Yamazaki, et al., 2021) is misled by the video content, suggesting an interval from the initiation of the first groundtruth action of piercing until the credit cut. In contrast, our proposed AOE-Net adeptly identifies intervals aligned with the actual groundtruth actions. This underscores the significance of both the actors beholder and objects beholder, which contribute more informative features compared to previous methodologies, thereby yielding superior results.

In the video depicting a challenging scenario (Fig. 2.10-A), the protagonists are hockey players, their scale minimized within the video frames. Consequently, discerning the initial "hockey playing" activity, juxtaposed with the subsequent "celebrating" activity, proves to be a formidable task due to the players’ diminutive appearance. The intricacy arises from the necessity to meticulously scrutinize the movements of the hockey players to distinguish between these activities. Both BMN and DBG, consequently, falter in recognizing the actual action interval as per the ground truth. In contrast, ABN can propose an interval encompassing the field scene but not necessarily aligning with the authentic action timeline. Conversely, our proposed AOE-Net adeptly suggests an interval closely mirroring the ground truth action, underscoring the significant contributions of our actor and object observation enhancements.

In the scenario involving non-human elements (Fig. 2.10-C), the actor engages in cooking activities, showcasing their hands manipulating a pan within the time range of [13.9-99.17]. Concurrently, the periods of [0.0-13.9] and [99.17-116.64] are designated for displaying advertisements. Due to the exclusive presentation of hands during the action in the video frames, the Actors Beholder fails to detect the actor. Nevertheless, our Objects Beholder effectively discerns the advertisement intervals at the video’s outset and conclusion, thereby accurately identifying the genuine action interval nestled in between. In contrast, the prior state-of-the-art (SOTA) model, BMN [3], erroneously interprets the advertisement breaks as authentic actions, either categorizing them as distinct action intervals or erroneously amalgamating them with the genuine action interval situated between them.

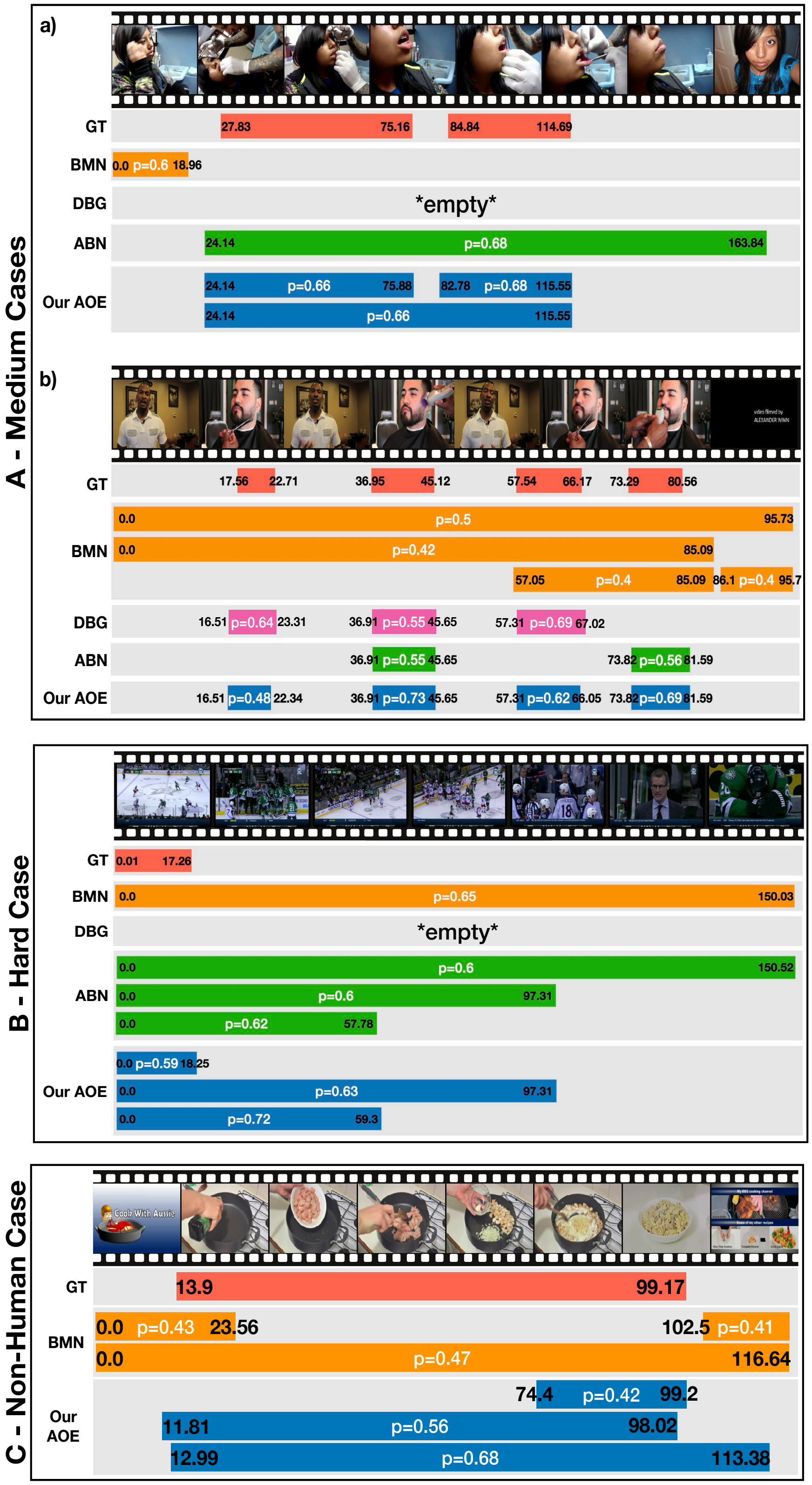


Figure 2.10: Results of a qualitative nature in the TAPG framework on the ActivityNet-1.3 dataset (Fabian Caba Heilbron and Niebles, 2015).

Chapter 3

CONCLUSION & FUTURE WORK

In this thesis, our objective is to simulate human perceiving abilities, and we introduce a novel AOE-Net designed to locate actions in untrimmed videos. The AOE-Net consists of two modules: PMR and BMM. PMR extracts visual-linguistic representations of each snippet with four beholders. The environment beholder and actors beholder capture global and local visual features of the environment and main actors, respectively. The objects beholder extracts linguistic features from relevant objects, while the last beholder models the relations between main actors, relevant objects, and the environment. To focus on an arbitrary number of main actor(s) or relevant objects, we introduce AAM. Qualitative and quantitative results on ActivityNet-1.3 and THUMOS-14 datasets for both TAPG and TAD tasks suggest that our proposed AOE-Net outperforms state-of-the-art methods. To demonstrate the effectiveness of AOE-Net, we provide ablation studies showcasing the contribution of each beholder, the effectiveness of the proposed AAM, network efficiency, and the robustness of AOE-Net when applied to egocentric videos from the EPIC-KITCHENS 100 dataset. We further investigate the performance of AOE-Net with various backbone network configurations, emphasizing that replicating human perceiving ability in video understanding holds promise for future exploration.

Several potential directions for future research stem from this work. Firstly, while main actors and relevant objects significantly impact both TAPG and TAD tasks, exploring the influence of human body parts (e.g., hands, legs) and their interactions with objects in localizing human activities in untrimmed videos would be of great interest. Finally, integrating our method with human tracking, specifically main actors tracking, may yield even better performance.

BIBLIOGRAPHY

Agarap, Abien Fred (2018). “Deep learning using rectified linear units (relu)”. In: *arXiv preprint arXiv:1803.08375*.

Bai, Yueran, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu (2020). “Boundary content graph neural network for temporal action proposal generation”. In: *ECCV*. Springer, pp. 121–137.

Bodla, Navaneeth, Bharat Singh, Rama Chellappa, and Larry S. Davis (Oct. 2017). “Soft-NMS – Improving Object Detection With One Line of Code”. In: *ICCV*.

Buch, Shyamal, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles (2017). “End-to-End, Single-Stream Temporal Action Detection in Untrimmed Videos”. In: *BMVC*.

Carreira, Joao and Andrew Zisserman (2017). “Quo vadis, action recognition? a new model and the kinetics dataset”. In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308.

Chao, Yu-Wei, Sudheendra Vijayanarasimhan, et al. (2018). “Rethinking the Faster R-CNN Architecture for Temporal Action Localization”. In: *CVPR*, pp. 1130–1139.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078*.

Dai, Xiyang, Bharat Singh, et al. (2017). “Temporal Context Network for Activity Localization in Videos”. In: *ICCV*, pp. 5727–5736.

Damen, Dima, Hazel Doughty, et al. (2021). “Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100”. In: *IJVC*, pp. 1–23.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *ICLR*.

Eun, H., S. Lee, J. Moon, J. Park, C. Jung, and C. Kim (2019). “SRG: Snippet Relatedness-based Temporal Action Proposal Generator”. In: *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1.

Fabian Caba Heilbron Victor Escorcia, Bernard Ghanem and Juan Carlos Niebles (2015). “ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding”. In: *CVPR*, pp. 961–970.

Feichtenhofer, Christoph, Haoqi Fan, et al. (2019). “SlowFast Networks for Video Recognition”. In: *ICCV*, pp. 6201–6210.

Feichtenhofer, Christoph, Haoqi Fan, Jitendra Malik, and Kaiming He (2018). “SlowFast networks for video recognition. 2019 IEEE”. In: *CVF international conference on computer vision (ICCV)*, pp. 6201–6210.

Gao, Jialin, Zhixiang Shi, Guanshuo Wang, Jiani Li, Yufeng Yuan, Shiming Ge, and Xi Zhou (2020). “Accurate temporal action proposal generation with relation-aware pyramid network”. In: *AAAI*. Vol. 34. 07, pp. 10810–10817.

Gao, Jiyang, Kan Chen, and Ram Nevatia (2018). “CTAP: Complementary Temporal Action Proposal Generation”. In: *ECCV*. Vol. 11206, pp. 70–85.

Gao, Jiyang, Runzhou Ge, Kan Chen, and Ram Nevatia (June 2018). “Motion-Appearance Co-Memory Networks for Video Question Answering”. In: *CVPR*.

Gao, Jiyang, Zhenheng Yang, et al. (2017). “TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals”. In: *ICCV*, pp. 3648–3656.

Gao, Jiyang, Zhenheng Yang, and Ram Nevatia (May 2017). “Cascaded Boundary Regression for Temporal Action Detection”. In: *arXiv e-prints*, arXiv:1705.01180, arXiv:1705.01180. arXiv: 1705.01180 [cs.CV].

Geva, Mor, Roei Schuster, Jonathan Berant, and Omer Levy (2020). “Transformer feed-forward layers are key-value memories”. In: *arXiv preprint arXiv:2012.14913*.

Ging, Simon, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox (2020). “COOT: Cooperative Hierarchical Transformer for Video-Text Representation Learning”. In: *Advances on Neural Information Processing Systems (NeurIPS)*.

He, Kaiming, Georgia Gkioxari, Piotr Dollar, and Ross Girshick (Oct. 2017). “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Heilbron, Fabian Caba, Juan Carlos Niebles, and Bernard Ghanem (2016). “Fast Temporal Activity Proposals for Efficient Detection of Human Actions in Untrimmed Videos”. In: *CVPR*, pp. 1914–1923. DOI: 10.1109/CVPR.2016.211.

Hendrycks, Dan and Kevin Gimpel (2016). “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv:1606.08415*.

Henighan, Tom, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. (2020). “Scaling laws for autoregressive generative modeling”. In: *arXiv preprint arXiv:2010.14701*.

Vo-Ho, Viet-Khoa, Ngan Le, Kashu Kamazaki, Akihiro Sugimoto, and Minh-Triet Tran (2021). “Agent-Environment Network for Temporal Action Proposal Generation”. In: *ICASSP*, pp. 2160–2164.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780.

Ji, Shuiwang, Wei Xu, Ming Yang, and Kai Yu (2013). “3D Convolutional Neural Networks for Human Action Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1, pp. 221–231. DOI: 10.1109/TPAMI.2012.59.

Jiang, Y.-G., J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar (2014). *THUMOS Challenge: Action Recognition with a Large Number of Classes*. `http://crcv.ucf.edu/THUMOS14/`.

Jozefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever (2015). "An empirical exploration of recurrent network architectures". In: *International conference on machine learning*. PMLR, pp. 2342–2350.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361*.

Kay, Will, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman (2017). "The Kinetics Human Action Video Dataset". In: *CoRR* abs/1705.06950. arXiv: `1705.06950`. URL: `http://arxiv.org/abs/1705.06950`.

Krishna, Ranjay, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles (2017). "Dense-Captioning Events in Videos". In: *International Conference on Computer Vision (ICCV)*.

Lin, Chuming, Jian Li, et al. (2020). "Fast Learning of Temporal Action Proposal via Dense Boundary Generator". In: *AAAI*, pp. 11499–11506.

Lin, Tianwei, Xiao Liu, et al. (2019). "BMN: Boundary-Matching Network for Temporal Action Proposal Generation". In: *ICCV*, pp. 3888–3897.

Lin, Tianwei, Xu Zhao, et al. (2018). "BSN: Boundary Sensitive Network for Temporal Action Proposal Generation". In: *ICCV*. Vol. 11208, pp. 3–21.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014). "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, pp. 740–755. ISBN: 978-3-319-10602-1.

Liu, Shuming, Xu Zhao, et al. (2020). "TSI: Temporal Scale Invariant Network for Action Proposal Generation". In: *ACCV*.

Liu, Yuan, Lin Ma, et al. (2019). "Multi-Granularity Generator for Temporal Action Proposal". In: *CVPR*, pp. 3604–3613.

Long, Fuchen, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei (2019). "Gaussian temporal awareness networks for action localization". In: *CVPR*, pp. 344–353.

Malinowski, Mateusz, Carl Doersch, Adam Santoro, and Peter Battaglia (2018). "Learning visual question answering by bootstrapping hard attention". In: *ECCV*, pp. 3–20.

Neubeck, A. and L. Van Gool (2006). "Efficient Non-Maximum Suppression". In: *ICPR*. Vol. 3, pp. 850–855.

Qing, Zhiwu, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang (2021). "Temporal Context Aggregation Network for Temporal Action Proposal Refinement". In: *CVPR*, pp. 485–494.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (18–24 Jul 2021). “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 8748–8763. URL: `https://proceedings.mlr.press/v139/radford21a.html`.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc.

Richard, Alexander and Juergen Gall (2016). “Temporal Action Detection Using a Statistical Language Model”. In: *CVPR*, pp. 3131–3140.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. DOI: `10.18653/v1/P16-1162`. URL: `https://aclanthology.org/P16-1162`.

Shou, Zheng, Dongang Wang, and Shih-Fu Chang (2016). “Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs”. In: *CVPR*, pp. 1049–1058. DOI: `10.1109/CVPR.2016.119`.

Simonyan, Karen and Andrew Zisserman (2014). “Two-Stream Convolutional Networks for Action Recognition in Videos”. In: *NIPS*, pp. 568–576.

Su, Haisheng, Weihao Gan, et al. (2021). “BSN++: Complementary Boundary Regressor with Scale-Balanced Relation Modeling for Temporal Action Proposal Generation”. In: *AAAI*, pp. 2602–2610.

Tan, Jing, Jiaqi Tang, Limin Wang, and Gangshan Wu (2021). “Relaxed transformer decoders for direct action proposal generation”. In: *ICCV*.

Tran, D, L Bourdev, R Fergus, L Torresani, and M Paluri (2014). “Learning Spatiotemporal Features with 3D Convolutional Networks. ArXiv e-prints”. In: *arXiv preprint arXiv:1412.0767*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017a). “Attention is All you Need”. In: *NeurIPS*. Curran Associates, Inc.

– (2017b). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Vo, Khoa, Hyekang Joo, Kashu Yamazaki, Sang Truong, Kris Kitani, Minh-Triet Tran, and Ngan Le (2021). “AEI: Actors-Environment Interaction with Adaptive Attention for Temporal Action Proposals Generation”. In: *32nd British Machine Vision Conference 2021, BMVC 2021, Virtual*

Event, UK, November 22-25, 2021. URL: https://www.bmvc2021-virtualconference.com/assets/papers/1095.pdf.

Vo, Khoa, Kashu Yamazaki, Sang Truong, Minh-Triet Tran, Akihiro Sugimoto, and Ngan Le (2021). "ABN: Agent-Aware Boundary Networks for Temporal Action Proposal Generation". In: *IEEE Access* 9, pp. 126431–126445.

Wang, Limin, Yuanjun Xiong, Dahua Lin, and Luc Van Gool (July 2017). "UntrimmedNets for Weakly Supervised Action Recognition and Detection". In: *CVPR*.

Wang, Xiang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang (2021). "Self-supervised learning for semi-supervised temporal action proposal". In: *CVPR*, pp. 1905–1914.

Xiong, Yuanjun, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang (2016). "CUHK & ETHZ & SIAT Submission to ActivityNet Challenge 2016". In: *CoRR* abs/1608.00797.

Xu, Mengmeng, Chen Zhao, et al. (2020). "G-TAD: Sub-Graph Localization for Temporal Action Detection". In: *CVPR*, pp. 10153–10162.

Yao, T., Y. Li, Z. Qiu, et al. (2017). "Msr asia msm at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and densecaptioning events in videos". In: *CVPRW*.

Zeng, Runhao, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan (2019). "Graph convolutional networks for temporal action localization". In: *ICCV*, pp. 7094–7103.

Zhao, Peisen, Lingxi Xie, et al. (2020a). "Bottom-Up Temporal Action Localization with Mutual Regularization". In: *ECCV*. Vol. 12353, pp. 539–555.

Zhao, Peisen, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian (2020b). "Bottom-up temporal action localization with mutual regularization". In: *ECCV*. Springer, pp. 539–555.

Zheng, Jingye, Dihu Chen, and Haifeng Hu (2021). "Boundary Adjusted Network Based on Cosine Similarity for Temporal Action Proposal Generation". In: *Neural Processing Letters*, pp. 1–16.

INDEX