UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Collaborative Dual-Stream Modeling for Video Understanding

by Xiaohan Wang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Prof. Yi Yang

University of Technology Sydney
Faculty of Engineering and Information Technology

February 2021

Certificate of Authorship/Originality

I certify that the work in this thesis has not been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are quoted in the thesis. This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed
prior to publication.

February 2021

ABSTRACT

Collaborative Dual-stream Modeling for Video Understanding

by

Xiaohan Wang

Most existing video recognition systems classify the input video to coarse-grained labels with single-stream architectures or combine multi-modal predictions by simple late fusion. However, real-world video applications usually require understanding complex human-object interactions and fine-grained content. It expects a video analysis system to be able to conduct meticulous reasoning. Besides, the urgent need for multi-modal alignment and communication among different models requires multi-stream video modeling, which is beyond single-stream architectures' capacity.

In this thesis, I argue that we should tackle video understanding with collaborative dual-stream modeling in several challenging scenarios. The interaction between the different information in videos can encourage the video understanding system to exploit the spatio-temporal relation. The idea has been applied to three tasks. First, for egocentric action recognition, symbiotic attention mechanism and interactive prototype learning scheme are developed to explore the relationship between the motion stream and appearance stream. Second, we design a T2VLAD framework for text-video retrieval to align the text stream and video stream. Third, for efficient video recognition, the communication between the lightweight model and heavyweight model is enabled by a parallel sampling network to sample more salient frames. Extensive experiments on popular video datasets demonstrate the effectiveness of the proposed approaches.

Dissertation directed by Professor Yi Yang

School of Computer Science

Acknowledgements

I would like to thank my principal supervisor Prof. Yi Yang. He motivates me to set a high standard for my research and gives considerate advice on both research and career planning. Without his continuous and invaluable support, the Ph.D. degree can not be attainable.

I would like to thank my co-supervisor Dr. Linchao Zhu. He introduced me to the video understanding area and taught me a lot of research skills. He gave inspirational advice on my research and helped me a lot on paper writing and presentations.

I want to thank my main collaborators, Mr. Yu Wu at UTS, Dr. Heng Wang at FaceBook AI, Dr. Ping Liu at Singapore A*STAR and Mr. Haitian Zeng at Baidu Research. Thank you all for discussing with me on interesting ideas and helping me with paper writing.

Thanks to all my colleagues and friends in UTS and Baidu Research.

Finally, I would like to thank my parents Mr. Xun Wang and Ms. Xuechun Wang for their unconditional love over so many years. I want to thank my girlfriend Ms. Yang Su for her love, support, and encouragement during the past eight years.

Xiaohan Wang

Sydney, Australia, 2021.

List of Publications

Journal Papers

J-1. **X. Wang**, L. Zhu, Y. Wu and Y. Yang, "Symbiotic Attention for Egocentric Action Recognition with Object-centric Alignment," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2020.3015894. (**Published**)

J-2. Y. Wu, L. Zhu, **X. Wang**, Y. Yang, F. Wu, "Learning to Anticipate Egocentric Actions by Imagination," in *IEEE Transactions on Image Processing*. (**Accepted**)

Conference Papers

C-1. **X. Wang**, Y. Wu, L. Zhu, Y.Yang, "Symbiotic Attention with Privileged Information for Egocentric Action Recognition," in *AAAI*, pp. 12249-12256, 2020. (**Published**)

C-2. **X. Wang**, Y. Wu, L. Zhu, Y.Yang, "Baidu-UTS Submission to the EPIC-Kitchens Action Recognition Challenge 2019," in *CVPR Workshop*, 2019. (**Published**)

C-3. **X. Wang**, L. Zhu, H. Wang, Y.Yang, "What Are You Cutting? Recognizing Active Objects in Egocentric Videos via Interactive Prototype Learning." (**Under Review**)

C-4. **X. Wang**, L. Zhu, Y.Yang, "T2VLAD: Global-Local Sequence Alignment for Text-Video Retrieval," (**Under Review**)

C-5. **X. Wang**, L. Zhu, P. Liu, Y.Yang, "Parallel Sampling Network: A Differentiable Framework with Context Relation Mining for Efficient Video Recognition." (**Under Review**)

C-6. H. Zeng, Y. Dai, X. Yu, **X. Wang**, Y.Yang, "Learning Deep Clustered Representation in Shape Space for for Non-rigid Structure-from-Motion." (**Under Review**)

Contents

List of Figures

Chapter 1

Introduction

In the past few years, we have witnessed a significant progress in tackling many computer vision problems, *e.g.*, image classification [56, 38, 43], detection [32, 31, 81, 37], segmentation [67, 7, 37]. In video analysis, with the emerging of deep convolutional neural networks and large-scale datasets [6, 34, 11, 13], the video recognition performance has been prominently boosted [85, 92, 97, 110, 80, 62]. Most existing methods focus on classifying videos to coarse-grained labels with single stream architectures [92, 97, 110, 80] or combine multi-modal predictions by simple late fusion [85, 6]. However, real-world video applications usually face with complex human-object interactions and fine-grained content. Single-stream models tend to mix all input information without precise reasoning. Besides, the urgent need for multi-modal alignment and communication among different models requires multi-stream video modeling, which is beyond the capacity of single-stream architectures. To address these problems, we need to go beyond the single-stream architecture and introduce more interactions among different features and empower the model to leverage the semantic relations in videos.

How can we enable meticulous reasoning for complex video understanding? In this thesis, we focus on collaborative dual-stream modeling between different semantic features or modalities. For example, we can decompose the fine-grained action labels to verb and noun, and design an interaction module between the verb feature and noun feature. We have explored the dual-stream modeling for video understanding in three directions: (1) the collaborative dual-stream modeling between motion

information and object information for egocentric video recognition; (2) the collaborative dual-stream modeling between the light-weight model and heavy-weight model for efficient video recognition; (3) the collaborative dual-stream modeling between text description and intrinsic video content for cross-modal video retrieval. In the following sections, I will explain the three directions for video understanding.

1.1 Egocentric Action Recognition

Egocentric videos have become popular on social media and have attracted increasingly more attention in computer vision since the introducing of datasets such as EGTEA [59], Charades-Ego [84], EPIC-KITCHENS [12, 11, 13]. Unlike third-person videos where actions usually happen at a distance, egocentric videos focus on person and object interactions at a closer look. Understanding egocentric videos requires to identify both the motion from the actor and the object that the actor interacts with. Recent egocentric video datasets [11, 13] are usually constructed by decomposing an action into a combination of a *verb* and a *noun*, where action recognition can be achieved by classifying the associated *verb* and *noun*. For instance, "cut potato" is divided into a verb "cut" and a noun "potato". Such a formulation helps to distinguish subtle semantic differences among actions.

Egocentric video classification focuses on domain-specific fine-grained action recognition, while the existing third-person datasets [49] are more generic and collected from various domains like sports and daily activities. In egocentric videos, the background scene is often similar among actions. For instance, "cutting carrot" and "peeling potato" can happen in the same kitchen scene. Hence, the usefulness of background scene information is limited in egocentric videos, making the recognition more challenging.

In Chapter 3, we propose to tackle egocentric action recognition by suppressing background distractors and enhancing action-relevant interactions. The existing ap-

proaches usually utilize two independent branches to recognize egocentric actions, *i.e.*, a verb branch and a noun branch. However, the mechanism to suppress distracting objects and exploit local human-object correlations is missing. To this end, we introduce two extra sources of information, i.e., the candidate objects' spatial location and their discriminative features, to enable concentration on the occurring interactions. We design a **S**ymbiotic **A**ttention with **O**bject-centric feature **A**lignment framework (SAOA) to provide meticulous reasoning between the actor and the environment. First, we introduce an object-centric feature alignment method to inject the local object features to the verb branch and noun branch. Second, we propose a symbiotic attention mechanism to encourage the mutual interaction between the two branches and select the most action-relevant candidates for classification. The framework benefits from the communication among the verb branch, the noun branch, and the local object information. Experiments based on different backbones and modalities demonstrate the effectiveness of our method.

In Chapter 4, we propose an end-to-end interactive prototype learning (IPL) framework to learn better active object representations by leveraging the motion cues from the actor. Egocentric video recognition is a challenging task that requires to identify both the actor motion and the active object that the actor interacts with. Recognizing the active object is particularly hard due to the cluttered background with distracting objects, the frequent field of view changes, severe occlusion, etc. To improve the active object classification, most existing methods use object detectors or human gaze information, which are either computationally expensive or require labor-intensive annotations. To avoid the cost of doing explicit object detection or gaze estimation, we first introduce a set of verb prototypes to disentangle active object features from distracting object features. Each prototype corresponds to a primary motion pattern of an egocentric action, offering a distinctive supervision signal for object feature learning. Second, we introduce two interactive operations to

fulfil the extraction of active object features, *i.e.*, *noun*-to-*verb* assignment and *verb*-to-*noun* selection. These operations are parameter-efficient and can learn judicious location-aware features on top of 3D CNN backbones.

1.2 Text-video Retrieval

Text-video retrieval is a challenging task that aims to search relevant video contents based on natural language descriptions. The key to this problem is to measure text-video similarities in a joint embedding space. Most existing methods only consider the global cross-modal similarity or incorporate the local details through complex matching and reasoning. In Chapter 6, we introduce an end-to-end text-video sequence alignment method to examine global-local cross-modal distances. The multi-modal sequences are aggregated with a T2VLAD encoding function, where a set of joint centers are shared between the text and video modalities. The shared centers could bridge the semantic gap between texts and videos, and at the same time, learn a text-aligned video representation on the fly. T2VLAD can be easily implemented based on the original NetVLAD, which can serve as a new baseline for future research. In experiments, we achieve consistent improvements on three standard text-video retrieval benchmarks and outperform the previous state-of-the-art by a clear margin.

1.3 Efficient video recognition

Sampling relevant frames is crucial for efficient video recognition. Existing methods either develop standalone hand-designed sampling strategies or learn a sequential selection policy. However, two challenges remain unsolved. First, standalone sampling strategies are heuristically crafted, and they are intrinsically non-adaptive to different video backbones. Second, sequentially selecting the next optimal frame ignores temporal relations among all video frames. The sequential selection process

also hinders the application of these video samplers in speed-critical systems. In Chapter 5, we propose a parallel video sampling network (PSN) under a differentiable framework to tackle the aforementioned issues. We optimize the video sampler with gradients from a differentiable surrogate loss, which allows dynamically altering the sampler with the cooperation from the video recognition model. Besides, we propose to model the inter-relation among contextual frames observed by the sampler, which encourages the sampler to select frames based on a comprehensive inspection of the entire video. We observe that a simple context relation mining instantiation would significantly improve the classification performance. The video sampler assesses a candidate set of frames swiftly and determines the significance of each frame in parallel. More than that, our sampler considers the feedback from all frames jointly, eliminating the learning difficulties of sequential decision making. The whole learning process is pure gradient-based, and therefore, the sampler can be learned in high efficiency.

Chapter 2

Literature Survey

I investigate egocentric action recognition, efficient video recognition, and text-video retrieval tasks in this thesis. In this chapter, I introduce the background and the related literature. I first introduce some recent works in deep video recognition in Section 2.1. The literature review on egocentric video recognition is presented in Section 2.2. Egocentric action recognition aims to identify the object that the human is interacting with. Thus, it is also related to the human-object interaction task. The related works are introduced in Section 2.3. In our egocentric video recognition framework, we design a symbiotic attention mechanism to enhance the interactions between verb branch and noun branch. So I introduce some related works about visual attention in Section 2.4. The typical methods for efficient video recognition and efficient neural network design are introduced in Section 2.5. I review the literature on the Text-Video retrieval task in Section 2.6. In Chapter 4 and Chapter 7, the proposed Interactive Prototype Learning method and the T2VLAD framework are related to the previous work VALD Encoding. I introduce some related works in section 2.7.

2.1 Deep Video Recognition

The application of deep neural networks has significantly advanced video recognition. Early works leverage 2D convolution networks to tackle video recognition task. Simonyan et al.developed a two-stream 2D CNN [85] using both RGB frames and optical flow as input. Donahue et al. [17] leveraged LSTM [40] to model the frame features extracted by 2D CNN. Wang et al.proposed TSN [97] to sample frames

from multiple temporal segments and aggregate all predictions for video recognition. Recently, TRN [122] and TSM [62] developed a temporal relation module and a temporal shift module to enhance the temporal modeling capability of 2D CNN, respectively.

Tran et al. [92] introduced 3D convolution for video recognition. I3D [6] initializes the 3D CNN with the inflated weights of 2D CNN pretrained on ImageNet [56]. Recently, S3D [110], R(2+1)D [94] and P3D [80] are proposed to decompose 3D convolution to a 2D spatial convolution and a 1D temporal convolution. Feichtenhofer et al.developed a SlowFast Network using the combination of two pathway 3D CNN with different temporal and spatial resolution. The design of these deep video recognition models does not consider the challenges in first-person video recognition. In Chapter 3 and Chapter 4, our method is based on the success of 3D CNNs on action recognition.

2.2 Egocentric Action Recognition

Compared to third-person action recognition, egocentric action recognition requires finer understanding of hand motion and the active objects in the complex environment. A number of existing methods leveraged object detection features to improve egocentric video recognition [98, 99, 103, 83, 69], among which [103, 83] also incorporate longer temporal contexts to support the understanding of the ongoing action. These methods require labor-intensive object detection annotations and prohibitive computation cost, which limits their application in real-world systems. In contrast, our framework only leverage the action labels as supervision and do not require additional data processing on high resolution frames.

Spatio-temporal attention is another way to locate the active object and anticipate the intention of the actor. Sudhakara et al. [89] proposed a two-stage Long Short Term Attention RNN model to track the discriminative area. It generates

attention maps based on the given feature map using a Class Activation Mapping (CAM)-like weighting, where a pre-trained object recognition network is required to serve as a prior knowledge to guide the localization of active regions from the feature map. Li et al. [59] and Liu et al. [65] leveraged the gaze annotations to guide deep models to focus on the interacting area and select informative features. Compared to these methods, our framework take advantage of the relation between motion information and active object to select useful features.

Moreover, a few works, e.g., TBN [50], leverage multi-modal inputs to enhance the recognition system. Multi-modal cues such as optical flow and audio are demonstrated to be effective in egocentric action recognition. In Chapter 3 and Chapter 4, we focus on learning the interactions between verbs and nouns to enhance egocentric representation learning.

2.3 Human-Object Interaction

Reasoning the interaction between human and objects is relevant to our task because it also requires to find out the interacting object. Most methods in this field are based on detection models. For example, Gkioxari et al. [33] predicted a density map to locate the interacted object and calculated the action score, with a modified Faster RCNN architecture. Qi et al. [79] proposed Graph Parsing Neural Networks that incorporates structural knowledge and deep object detection model. Fang et al. [22] developed a pairwise body-part attention model that can learn to focus on crucial parts for human-object interaction (HOI) recognition. Besides, some works use human-object interactions to help recognize actions. Wang et al. [101] proposed to represent videos as space-time region graphs, which models shape dynamics and relationships between actors and objects. Sun et al. [91] developed an Actor-Centric Relation Network for spatio-temporal action localization.

Most of these HOI techniques rely on the appearance of the actors, which is

absent in egocentric videos. Instead of the use of the detection features of humans, we pay attention to the interactions between the motion and the objects in Chapter 3 and Chapter 4.

2.4 Visual Attention

Attention mechanism can highlight visual regions or linguistic words that are important to the task predictions. It has been widely used in both computer vision [96, 42, 100, 64, 105, 1] and natural language processing [63, 95, 113]. Non-local networks [100] leveraged the non-local attention operation in spatio-temporal dimension for video recognition. Squeeze-and-excitation network (SENet) [42] developed the squeeze-and-excitation block, which introduced the channel-wise attention inside the residual block. Recently, Linsley et al. [64] improved the SE module by the global-and-local attention (GALA), which combined global contextual guidance with local saliency. In addition, they also introduced a large-scale dataset containing human-derived attention maps, which can be used to supervise the attention mechanism to be more accurate and interpretable. These methods are designed with a self-attention operation. The feature map used to generate attention weights is also the target feature where the weights are applied. Differently, our work concentrates on the interactions in egocentric videos. We apply a cross attention mechanism between the motion feature and the object appearance feature.

2.5 Efficient Computing

The computation complexity of CNNs limits their application in the real world. To overcome the limitation, some researchers developed lightweight and efficient networks such as MobileNet [41] and ShuffleNet [70]. These lightweight networks can serve as the backbone of the sampler in our framework, which can improve the efficiency of our approach. [10] propose to select the filters in CNN to reduce

computational cost. [87] develop a spatial sampling method for the CNN input. The idea of dynamic selection is similar to ours. But we focus on the selection of temporal frames for the networks. For efficient video recognition, [126] propose to capture motion information between frames rather than pre-computing optical flow towards real-time video recognition. [4] propose to teach the model to recognize video using fewer frames by knowledge distillation. [93] apply channel-wise convolution operation in 3D CNN. In Chapter 5, our framework improves the efficiency of video recognition by only processing the most informative part in videos.

Most of the existing works utilize hand-designed strategies for video sampling. [97] propose to divide an input video into several segments and randomly sample one frame from each segment. [6] utilize successive frame sampling to train a 3D CNN. And uniform sampling is widely used in the testing scenario. Recently, [53] propose to select the most salient clips for testing based on their classification score of a lightweight net. The most relevant works with our method are learning-based sampling strategies [115, 127, 21, 104, 108, 107]. [115, 21] propose to learn where to look and whether to stop in a video by reinforcement learning. [108] introduce an adaptive strategy and a global memory for frame sampling. [104] utilize a multi-agent reinforcement learning algorithm to train a context-aware sampler. The above approaches for learnable sampling strategies are all based on the reinforcement learning framework except [107]. Our method introduces a gradient-based training scheme. Besides, these previous strategies sample frames in a sequential manner. In contrast, our method samples candidates in parallel, which is more efficient in practice.

2.6 Text-Video Retrieval

There are increasing interests in advancing text-video retrieval performance [77]. Compared to text-image retrieval [20, 51, 48], text-video retrieval is more challenging that requires the understanding of temporal dynamics and complicated text

semantics. A few works [77, 76] focus on visual semantic embedding learning for text and video joint modeling. Mithun et al. [76] leveraged a simple text-image embedding method [20] to improve the training strategy with hard negative mining, and incorporated multi-modal features (RBG, motion, and audio) to enrich the video representations. Dong et al. [19] proposed dual-encoding network with multiple levels of features for text-video retrieval, *i.e.*, features obtained by mean pooling, bi-directional Gated Recurrent Unit and Convolution Neural Networks. Yu et al. [116] proposed a joint fusion model using Long Short-Term Memory for temporal sequential information encoding between videos and texts. Liu et al. [66] further utilize all modalities that can be extracted from videos such as speech contents and scene texts for video encoding. Miech et al. [74] introduced a strong joint embedding using mixture-of-expert features, which are later utilized in [26].

2.7 VLAD Encoding

VLAD [46] and NetVLAD [2] have achieved great impacts in aggregating discriminative features for video classification [30, 114], video retrieval [74], person re-identification [119]. NetVLAD is an end-to-end differentiable layer that could be readily plugged into many existing convolutional networks These works usually leverage the NetVLAD layer as a discriminative feature learner for downstream tasks. However, in Chapter 6, we leverage NetVLAD in text-video local similarity matching and introduce a local alignment loss to reduce the gap of locally learned features from texts and videos. In Chapter 6, we do not conduct classification upon the obtained aggregated features, but apply local alignment between the text and video features.

Chapter 3

Symbiotic Attention for Egocentric Action Recognition

3.1 Introduction

Egocentric video recognition is an important task in the video understanding field. It is valuable for practical applications such as human-computer interaction, intelligent wearable devices, and service robots. In this Chapter, we design a collaborative dual-stream modeling framework for egocentric action recognition. Most existing methods focus on recognizing videos captured from a third-person viewpoint. The progress in the first-person video has been relatively slow. Recently, egocentric action recognition has attracted increasing attention with the widespread applications of wearable cameras.

Compared to third-person videos, egocentric videos contain more complex scenes. Egocentric action recognition requires to distinguish the object that human is interacting with from various small distracting objects [14, 12]. Action recognition in egocentric videos provides a uniquely naturalistic insight into how a person or an agent interacts with the world. To enable the recognition of more complex videos, a challenging large-scale first-person dataset, *i.e.*, EPIC-Kitchens [12], was recently introduced for egocentric daily human activities understanding. This dataset provides rich interactions, covering adequate objects and natural actions. The intense camera motion, occlusion, and first-person viewpoint make it even more challenging to recognize fine actions.

In EPIC-Kitchens, the actions are defined by the combination of verb and noun,

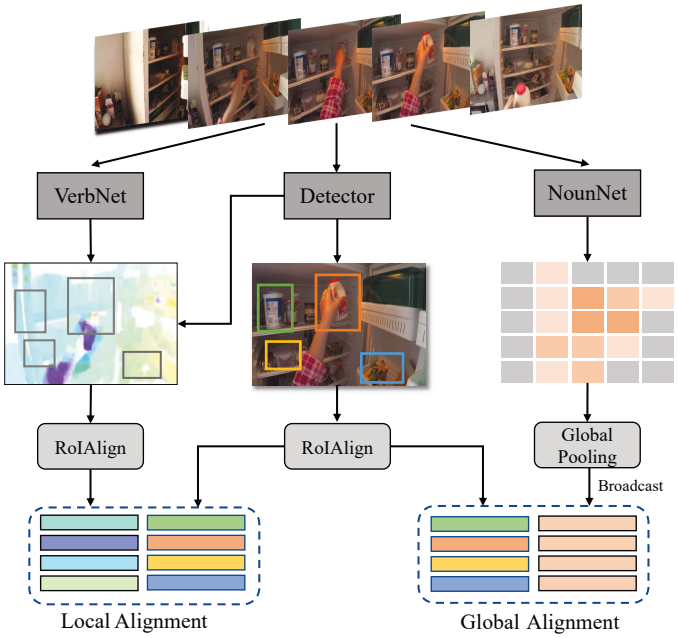

Figure 3.1 : The illustration of the object-centric feature alignment. For verb classification, the spatial location provided by the detector can possible reduce the object-irrelevant motions. Local motion features aligned with object features serve as possibly action candidates. For noun classification, global alignment inject the local object features into the context-aware global feature. These location-aware feature candidates from the two branches are beneficial to the subsequent meticulous reasoning.

e.g., "open door" and "cut potato". Due to the large action vocabulary, the verb and the noun classifiers are usually trained separately [12, 103]. The verb branch focuses on classifying actions (verbs) that the actor is performing, *e.g.*, "cut" and "open". The main obstacles for verb classification are large camera motion and subtle occurring action locations. The noun branch is to identify the object that the actor is interacting with. As shown in Figure 3.1, distracting objects in oblique view decrease the prediction score of the interacting object.

Damen et al. [11] evaluated several video models on EPIC-Kitchens that were not

specially designed for the egocentric action recognition such as TSN [97], TSM [62] and TRN [122]. These models failed to achieve high classification score due to the absence of location-aware guidance for the complex scenes in first-view videos.

Recently, Wu et al. [103] leveraged object detection features to introduce longer context information for the noun classification network in egocentric action recognition. The long-term feature bank is aggregated via a simple max pooling or average pooling operation, while the more sophisticated non-local operator is found to be not that effective. The verb branch and the noun branch are optimized independently. They only consider the interaction between the noun branch and the object features but fail to enable the communication between the verb branch and the noun branch. Baradel et al. [3] designed an object relation network for high-level object reasoning, where the relation modeling branch facilitates object masks to generate local object features. The object reasoning only performs on the object branch. It lacks interactions with the activity branch. These works ignore the mutual communication between the standalone verb and noun branches. They only focus on contextual modeling and relation reasoning on a *single* branch. However, an action is determined by both the interacting object and the motion that the actor is performing. It could be difficult even for a human to recognize an action by only looking at the objects while ignoring the actor's intention, or only understanding motion changes without the awareness of the interacting object.

To better exploit the local object guidance and the mutual benefits of the interactions between different branches, we make the following contributions.

We first propose an object-centric feature alignment method to dynamically integrate location-aware information to the verb and the noun branches. Our object-centric feature alignment encourages the meticulous reasoning between the actor and the environment. The object-centric features are extracted by an object detec-

tion model, providing finer local information that is beneficial to the attendance of an on-going action. The noun branch and the verb branch integrate location-aware information by two different approaches (Fig. 3.1). We introduce **global alignment for noun classification.** The noun features and the detection features are complementary to each other, and proper integration of these two features produces more accurate identification of the interacted object. In this global alignment, we concatenate each detection feature with the global noun feature. The generated features incorporate both local relevant features and global contextual features, which restrain the features of irrelevant objects. We introduce **local alignment for verb classification.** The verb feature contains motion information, which is quite different from the appearance information in noun feature and object features. The semantic gap between verb features and detection features is larger than the gap between noun features and detection features. It may not be straightforward to integrate global verb features with detection features directly. When we use the aforementioned global alignment for verb classification, it may generate indistinct features due to the accompanying background motion noises. We propose to integrate spatially-aligned verb features with object features. In this way, the most relevant verb features will be generated for better alignment with local object features. It eases the difficulties of the integration between verb features and local object features. We extract regional verb features from the verb branch by pooling from the spatial feature map with the given candidate spatial location. The regional motion feature is then combined with the corresponding detection feature.

After the object-centric alignment, we obtain a set of candidate verb features and noun features. A symbiotic attention mechanism is then introduced to enable mutual interactions between the two branches and select the most action-relevant features. It consists of two parts, *i.e.*, cross-stream gating mechanism and action-attended relation module. The fused object-centric features contain useful local de-

tails. However, due to the existence of inaccurate detection regions, there are quite a few disturbing background noises in the features. To this end, we propose a cross-stream gating mechanism to normalize the aligned features. This normalization process suppresses the action-irrelevant noises and enables mutual communication between the verb branch and the noun branch. To further uncover the relationships among the object-centric features and identify the most action-relevant information, we develop an action-attended relation module to examine each potential motion-object pair and then generate the final representation for classification. The proposed Symbiotic Attention with Object-centric Alignment (SAOA) method dynamically integrates three sources of information towards better action recognition.

We evaluate our framework with different backbones and modalities on the largest egocentric video dataset, *i.e.*, EPIC-Kitchens. We conducted experiments on two backbones (*i.e.*, I3D [6] and ResNet-50 [38]), two modalities (*i.e.*, RGB and optical flow). It can consistently improve the performance over the baselines by a large margin with different backbone and input modalities. The effectiveness of our framework is validated both quantitatively and qualitatively. Notably, our method outperforms the state-of-the-art method [50] by **6.7%** on the unseen test set and **2.9%** on the seen test set of Epic-Kitchens. The ensemble of the proposed method achieved first place in EPIC-Kitchens Action Recognition Challenge 2020.

In our previous work [98], global alignment is developed to integrate the object features for both the verb branch and the noun branch. And only the model with RGB frames as input and ResNet-3D [35] backbone were studied. We extend [98] by proposing a local alignment for verb classification to reduce the object-irrelevant motions and alleviate the large semantic gap between object features and motion features. Moreover, extensive experiments on different backbone and input modalities are conducted. That demonstrates our method is general, and the two-stream SAOA can significantly improve the recognition performance over the previous model SAP.

In summary, our main contributions are as follows:

First, we develop an object-centric alignment method to inject local details into the verb branch and noun branch. The alignment allows the model to take advantage of the location-aware object information and prevent it from confusing with background noise.

Second, we propose a novel symbiotic attention mechanism to enable the mutual interaction between the verb branch and the noun branch. It provides the meticulous reasoning between the actor and the environment. The experiment shows that the symbiotic attention is beneficial to distinguish the action-relevant motion and object.

Third, extensive experiments demonstrate the effectiveness and superiority of the proposed SAOA. Our results outperform the state-of-the-art by a large margin on the largest egocentric video dataset.

3.2 Method

3.2.1 Overview

In this section, we illustrate our network architecture for egocentric video recognition. We develop three backbone networks to extract features from the input video: (1) VerbNet is a 3D CNN and takes a video clip as input. It is designed to capture the motion information. (2) NounNet shares the same architecture with VerbNet. It is trained to produce a feature that represents object appearance. (3) Object detection model takes sampled individual frames as input. We use Faster R-CNN [81] as our detector to generate object features and location proposals. The output features and location proposals of the three base models are fed to the subsequent SAOA module. We aim to enable effective communication among VerbNet, NounNet, and object features. The SAOA module generates two feature vectors,

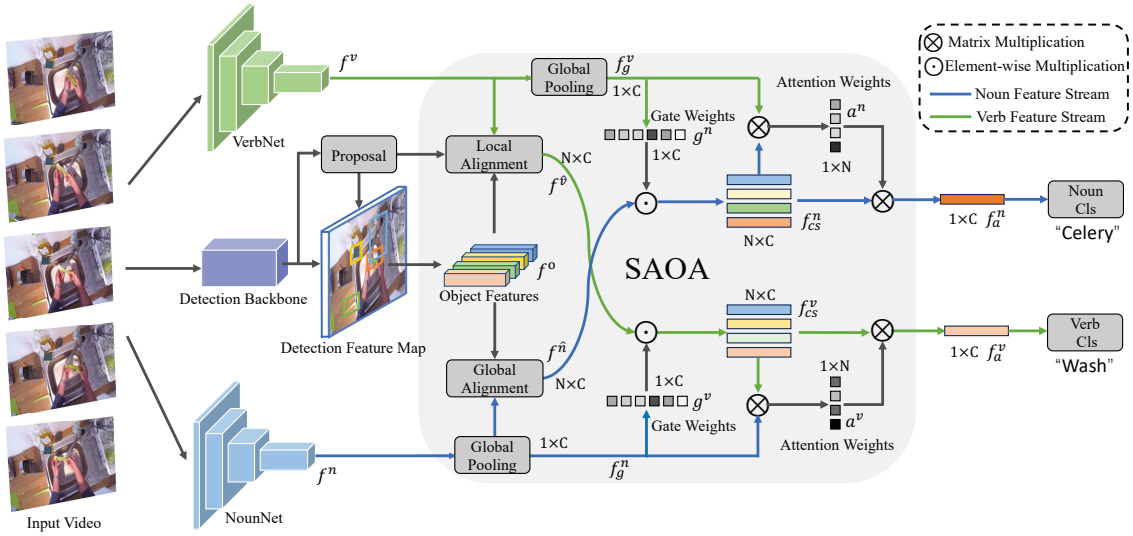

Figure 3.2 : The proposed SAOA framework. Our framework consists of three feature extractors and one interaction module. The detection model generates a set of local object features and location proposals. This location-aware information is injected to the two branches by an object-centric alignment method For the Verb branch, the feature map is locally aligned with the objects by combining the local motion features with corresponding object detection features. For the Noun branch, the object features are aligned with the global noun representation. Subsequently, the fused features from each branch interact with the global feature from the other branch by a symbiotic attention mechanism. The two object-centric feature matrices are first normalized by a cross-stream gating operation. After that, the matrices are attended by the other branch to select the most action-relevant information. The outputs of SAOA are used to classify the verb and noun, respectively.

which can be used to predict verb class and noun class. The overall framework is illustrated in Fig. 3.2.

3.2.2 Preliminaries

For each input egocentric video $X = \{x^1, ..., x^t\}$ with t frames, its verb and noun label is y^v and y^n, respectively. The action $y = (y^v, y^n)$ is a combination of the verb and noun. We use two individual 3D CNNs as the backbones in our framework, with one for the verb feature extraction and the other for the noun feature extraction. The extracted verb feature $f^v \in \mathbb{R}^{T \times H \times W \times C}$ contains the motion information, where T is the temporal size, H is the height, W is the width, and C is the channel size of the extracted feature, respectively. The noun feature $f^n \in \mathbb{R}^{T \times H \times W \times C}$ contains the global appearance information.

To enhance the global representation through the communication between two branches and enable meticulous reasoning, we use a pre-trained detection model to provide detailed locations of objects in the video. Considering the efficiency, for each video clip, we only use M sampled frames for detection inference. These frames are sampled around the center of the input clip for 3D CNNs within a fixed time duration. The duration is longer than the input clip to provide more context information. Given a feature map and a spatial location, $RoIAlign$ [36] first crops the feature map based on the location and then performs pooling operation to produce a fix-size feature map. In this work, we use max-pooling in $RoIAlign$ layer to produce a 1D feature vector. For object detection model, the output of the $RoIAlign$ layer is regarded as the feature for each detected object. To save memory usage and reduce the noisy information, we only keep top-K object proposals according to their confidence scores for each sampled frame. Thus, for each input clip of the 3D CNNs, we have a auxiliary object feature matrix $f^o \in \mathbb{R}^{N \times C_1}$, which contains $N = M \times K$ object features around the center of the short video clip. For each object detection feature $f_i^o, i \in [1 \dots N]$, we have an spatial detection location $l_i \in \mathbb{R}^4$. $l_i = (x_i^0, y_i^0, x_i^1, y_i^1)$ representing a rectangular in 2D space. The object feature matrices f^o are fused with the verb feature f^v and noun feature f^n by the following object-centric

alignment method. After that, the verb branch and noun branch are interacted with each other to produce more discriminative features for action recognition with a symbiotic attention mechanism.

3.2.3 Object-centric Feature Alignment

The verb branch and noun branch produce two feature maps f^v and f^n by passing a video clip to each backbone. Due to the intensive camera motion and various distracting objects in egocentric videos, the useful interaction information in these features is hard to distinguish from the global feature map without any other guidance. To this end, we develop an object-centric feature alignment method to generate potential motion and object candidates, which disentangle the local information from the global feature maps. Specifically, we leverage object feature matrix f^o and corresponding locations as location-aware information to inject the local details into the global features. Considering the different semantic properties of the verb branch and noun branch, we introduce two mechanisms to integrate f^o with f^v and f^n:

Global alignment for the noun branch. The object features and the noun feature both represent the appearance of the objects in the videos. Considering the small semantic gap and the complementarity between the local object features and the global noun feature, we introduce a direct global alignment for the noun branch. Note that we have $f^o \in \mathbb{R}^{N \times C_1}$ and $f^n \in \mathbb{R}^{T \times H \times W \times C}$. We first leverage a global average pooling (GAP) on f^n, and the generated global feature vector f_g^n is of shape $1 \times C$. Each detection feature in f^o is then concatenated with the global feature vector, followed by a nonlinear activation. Formally, the global alignment operation can be presented as follow:

$$f_i^{\hat{n}} = \texttt{ReLU}(W^n f_g^{n\mathrm{T}} + W_o^n f_i^{o\mathrm{T}} + b^n), i \in [1 \dots N], \tag{3.1}$$

where $W^n \in \mathbb{R}^{C \times C}, W_o^n \in \mathbb{R}^{C \times C_1}, b_n \in \mathbb{R}^{1 \times C}$, f_i^o denotes i-th detection feature in f^o

and $f_i^{\hat{n}}$ is the aligned noun feature. We obtain the final noun feature $f^{\hat{n}} \in \mathbb{R}^{N\times C}$ by concatenating all $f_i^{\hat{n}}$ where $i \in [1 \dots N]$. Each row in $f^{\hat{n}}$ represent a object-centric feature, which integrates the global noun appearance with an explicit local object information.

Local alignment for the verb branch. Different from global alignment for the noun branch, we leverage a local alignment that integrates the verb feature map and the object detection features based on their spatial locations. The verb feature represents the motion information in the videos, which is quite different from the object features. Global alignment might not well integrate the two features due to the large semantic gap. The proposed local alignment can decompose the global motion information to object-centric local details. Note that we have $f^o \in \mathbb{R}^{N\times C_1}$ and $f^v \in \mathbb{R}^{T\times H\times W\times C}$. For each object detection feature $f_i^o, i \in [1 \dots N]$, we have a spatial detection location $l_i \in \mathbb{R}^4$. $l_i = (x_i^0,\ y_i^0,\ x_i^1,\ y_i^1)$ representing a rectangular in 2D space. We extract the locally aligned verb feature from f_i^v by the ROIAlign operation, *i.e.*, $f_i^v = ROIAlign(f^v, l_i)$. The final verb feature can be obtained via:

$$f_i^{\hat{v}} = \texttt{ReLU}(W^v f_i^{v\mathrm{T}} + W_o^v f_i^{o\mathrm{T}} + b^v), i \in [1 \dots N], \tag{3.2}$$

where $W^v \in \mathbb{R}^{C\times C}, W_o^v \in \mathbb{R}^{C\times C_1}, b_v \in \mathbb{R}^{1\times C}$ and $f_i^{\hat{v}}$ is the aligned verb feature. The final verb feature $f^{\hat{v}} \in \mathbb{R}^{N\times C}$ is obtained by concatenating all $f_i^{\hat{v}}$ where $i \in [1 \dots N]$. The final motion-object paired feature incorporates local detection features and location-aware motion features.

3.2.4 Symbiotic Attention

The object-centric alignment integrates the object features to the verb branch and noun branch. The fused object-centric feature matrices contain useful local details and provide potential action-relevant candidates for verb and noun classification. We propose a symbiotic attention mechanism to encourage mutual communication between the two branches. It further generates a better representation

for classification. As illustrated in Fig. 3.2, symbiotic attention includes two stages. First, the fused object-centric features are re-calibrated by the other branch utilizing a cross-stream gating mechanism. After that, the normalized feature matrix is attended by the other branch to aggregate the most action-relevant information within an action-attended relation module.

Cross-Stream Gating

Due to the existence of inaccurate detection regions, there are quite a few disturbing background noises in the features. Besides, it is important to introduce information from one branch to guide discrimination in the other branch. For example, given a video clip that presents the action "cut potato" but also contains the object "bowl", the motion information of "cut" can provide extra guidance for more accurate recognition that the interacted object is "potato" rather than "bowl". To this end, we develop a cross-stream gating operation to underline the action-relevant information from the verb branch and the noun branch.

In noun classification, we generate a gating weight to normalize the input noun feature matrix $f^{\hat{n}}$. The gating weight g^n is obtained from the global verb feature:

$$f_g^v = GAP(f^v), \tag{3.3}$$

$$g^n = \texttt{Sigmoid}(W_g^n f_g^{v\mathrm{T}} + b_g), \tag{3.4}$$

$$f_{cs}^n = g^n \odot f^{\hat{n}}, \tag{3.5}$$

where $W_g^n \in \mathbb{R}^{C\times C}$, $f_g^v \in \mathbb{R}^{1\times C}$, $b_g \in \mathbb{R}^{1\times C}$, $g^n \in \mathbb{R}^{1\times C}$, $f_{cs}^n \in \mathbb{R}^{N\times C}$, and $\odot$ denotes the element-wise multiplication. g^n is the scaling vector to rescale the noun feature matrix. After re-calibrating the object-centric noun feature by the verb feature, the distracting noises can be suppressed while the action-relevant channels can be

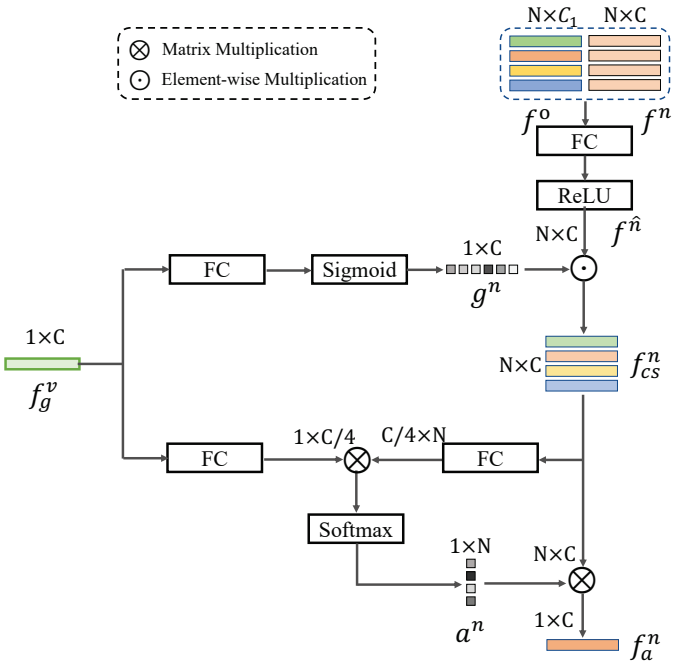

Figure 3.3 : The illustration of symbiotic attention on the noun branch. The object-centric noun feature matrix is first normalized by the global verb feature. After that, the feature matrix interacts with the global verb feature to generate attention weights. The final noun representation is the weighted sum of the normalized object-centric features.

enhanced. Similarly, the gated verb feature f_{cs}^{v} can be obtained by:

$$f_{g}^{n} = GAP(f^{n}), \tag{3.6}$$

$$g^{v} = \texttt{Sigmoid}(W_{g}^{v} f_{g}^{n\mathrm{T}} + b_{g}^{v}), \tag{3.7}$$

$$f_{cs}^{v} = g^{v} \odot f^{\hat{v}}, \tag{3.8}$$

where $W_{g}^{v} \in \mathbb{R}^{C\times C}$, $f_{g}^{n} \in \mathbb{R}^{1\times C}$, $b_{g}^{v} \in \mathbb{R}^{1\times C}$, $g^{v} \in \mathbb{R}^{1\times C}$, $f_{cs}^{v} \in \mathbb{R}^{N\times C}$. Our cross-stream gating mechanism enables mutual communication between the verb branch and the noun branch, and it adaptively exploits the correlations of verbs and nouns. We illustrate on the noun branch in Fig. 3.3.

Action-attended Relation Module

The calibrated object-centric feature matrix contains the action-relevant information and implicit guidance about the spatio-temporal location of an on-going action. To uncover the relationships among the object-centric features and identify the most action-relevant information, more meticulous reasoning is required. Therefore, we develop an action-attended relation module to examine each potential motion-object pair and then generate the final representation for classification.

Specifically, we first propose to assess the relevance between the global feature and location-aware object-centric features. Taking the noun branch for example. The global verb feature and the object-centric noun features are projected to the same dimension space. The distance between each noun feature and the verb feature is calculated to represent their relevance score. After that, we sum the object-centric features weighted by the relevance coefficients. Formally, we perform attention mechanism on the normalized object-centric noun features $f_{cs}^{n} \in \mathbb{R}^{N \times C}$ and the global verb feature $f_{g}^{v} \in \mathbb{R}^{1 \times C}$,

$$a^{n} = \texttt{Softmax}(f_{g}^{v} W_{v}^{a} W_{cn}^{a} f_{cs}^{n\,\mathrm{T}}), \tag{3.9}$$

where $W_{v}^{a} \in \mathbb{R}^{C \times \frac{C}{4}}$, $W_{cn}^{a} \in \mathbb{R}^{\frac{C}{4} \times C}$ are projection matrices. We project the features to a low feature dimension $\frac{C}{4}$ to reduce the computational cost of matrix multiplication. We found $\frac{C}{4}$ performs well in our experiments. $a^{n} \in \mathbb{R}^{1 \times N}$ is the generated attention weights. The final noun representation f_{a}^{n} is produced by the weighted sum of the object-centric features,

$$f_{a}^{n} = a^{n} f_{cs}^{n}. \tag{3.10}$$

Similarly, we select relevant action features from f_{cs}^{v} with query f_{g}^{n},

$$f_{a}^{v} = \texttt{Softmax}(f_{g}^{n} W_{n}^{a} W_{cv}^{a} f_{cs}^{v\,\mathrm{T}}) f_{cs}^{v}. \tag{3.11}$$

The final noun feature $f_a^n \in \mathbb{R}^{1\times C}$ and the final verb feature $f_a^v \in \mathbb{R}^{1\times C}$. Through the interaction of global feature and object-centric features, our model selects the most action-relevant feature for classification.

3.2.5 Training and Objectives

We use Faster R-CNN with the ResNeXt-101-FPN backbone as our object detector. Following the training procedure in [103], we first pre-train the detector on Visual Genome [55] and then finetune it on EPIC-Kitchens object detection set. For VerbNet and NounNet, we adopt 3D Resnet-50 [35] and I3D [6] as our backbones. The two networks are both initialized with Kinetics pre-trained weights. In the first stage, we individually train the VerbNet and NounNet with the corresponding CrossEntropy Loss, *i.e.*, $\mathcal{L}^v$ and $\mathcal{L}^n$.

$$\mathcal{L}^n = CrossEntropy(f_a^n, y^n), \tag{3.12}$$

$$\mathcal{L}^v = CrossEntropy(f_a^v, y^v). \tag{3.13}$$

After the base training stage, we freeze the weights of the backbone and cascade our SAOA module. The objective for the second stage is the same as the base training stage, and only the weights of SAOA are optimized.

3.3 Experiments

3.3.1 Datasets

EPIC-Kitchens is the largest dataset in first-person vision so far. It consists of 55 hours of recordings capturing all daily activities in the kitchens. The performed activities are non-scripted, which makes the dataset very challenging and close to real-world data. The dataset contains 39,594 action segments which are annotated with 125 verb classes (*e.g.*,"cut","take") and 321 noun classes (*e.g.*,"potato","knife").

The action of each video segment is defined by the verb-noun pair (*e.g.*, "cut potato", "take knife"). We split the original training set to new training and validation set following [3]. All hyper-parameters are selected based on the performance on the validation set. We report the top-1 and top-5 accuracy of the verb, noun, and action.

3.3.2 Experiment Settings

We implement and test our method using PaddlePaddle and PyTorch. We train our framework in a two-stage optimization scheme. Specifically, we firstly pre-train the base models (VerbNet, NounNet, and the detector) individually. After that, we optimize the subsequent SAOA module using extracted features from the base models. Next, we illustrate the details on how to pre-train the backbones (Backbone details) and how to extract local object information (Detector details). Finally, we show the training details of the module (SAOA details).

Backbone details. We adopt two typical 3D CNNs as our backbones, *i.e.*, ResNet50-3D [35] and I3D [6]. ResNet50-3D is built with residual blocks and I3D is based on Inception architecture. We take the Kinetics [6] pre-trained weights as the initialization of our backbone model. We then train the backbone models (VerbNet and NounNet) individually on the target dataset using 64-frame input clips. The targets for the VerbNet and NounNet are the verb label and noun label, respectively. The videos are decoded at 60 FPS for the EPIC-Kitchens dataset. We adopt the stochastic gradient descent (SGD) with momentum 0.9 and weight decay 0.0001 to optimize the parameters for 35 epochs. The overall learning rate is initialized to 0.003, and then it is changed to 0.0003 in the last 5 epochs. The batch size is 32. During the first training stage, the input frame size is 224×224, and the input frame is randomly cropped from a random scaled video whose side is randomly sampled in [224, 288]. We sample 64 successive frames with stride=2 from each segment to constitute the input clip. The center index of the input clip is randomly chosen in

the segment during training. For the testing, we sample a center clip per segment. We resize the clip to the size of 256×256 and use a single center crop of 224×224.

Detector details. Following [103], we use the same Faster R-CNN to detect objects and extract object features. The detector is first pre-trained on Visual Genome [55] and then fine-tuned on the training split of the EPIC-Kitchens dataset. We use a batch size of 12 and train the model on EPIC-Kitchens for 180k iterations for the trainval/test split. We use an initial learning rate of 0.005, which is decreased by a factor of 10 at iteration 140k and 160k. For the train/val split, we train the model for 150k iterations, and the learning rate decays at iteration 116k and 133k. Finally, our object features are extracted using *RoIAlign* from the detector's feature maps. For each video clip, we perform object detection on a set of frames that are sampled around the clip center within a fixed time duration. The time duration is set to 6 seconds for global alignment and 4 seconds for local alignment. The sample rate is at two frames per second. For each frame, we keep the top five features and proposals according to the confidence scores. Therefore, given a video clip, we obtain 60 detection features during global alignment. In local alignment, we obtain 40 detection features and corresponding locations.

SAOA details. We leverage the pre-trained backbone models and the detection models as the feature extractors. During the second-stage training, only the weights of SAOA are updated. We use SGD with momentum 0.9 and weight decay 0.0001 to optimize the parameters with batch-size of 32. For the model equipped with the I3D backbone, we train the model for 15 epochs. The learning rate is initialized to 0.001 and then reduced to 0.0001 in the last 5 epochs. For the models based on R-50, we train the model for 15 epochs, and the learning rate is set to a constant value 0.0001. Notably, since the detection features have different scales from the I3D features, the features from the I3D backbone need to be normalized before concatenation with detection features in the alignment modules. However, the feature from the R-50

backbone can be directly fed to the SAOA module without normalization. The main reason is the different network types between the detection backbones (based on residual block) and the I3D model (based on Inception block). Specifically, the features produced by the I3D backbone and detection model are l_2-normalized before concatenation. The combined feature is then multiplied by the l_2-norm of the I3D feature to scale the amplitude. A similar normalization strategy is introduced in [91]. During the training and testing of SAOA, we utilize the same temporal sampling strategy during the training and testing of the backbone. For each input video clip, we resize it to the size of 256. Then we feed the 64-frame clip to the network without spatial cropping.

Action calculation The actions are determined by the pairs of verb and noun. The basic method of obtaining the action score is to calculate the multiplication of verb probability and noun probability. However, there are thousands of combinations and most verb-noun pairs that do not exist in reality, *e.g.*, "open the knife". In fact, there are only 149 action classes that have more than 50 samples in the EPIC-Kitchens dataset [12]. Following the approach in [103], we re-weight the final action probability by a prior, *i.e.*

$$P(action = y) = \mu(y^v, y^n)P(verb = y^v)P(noun = y^n), \tag{3.14}$$

where μ is the occurrence frequency of action in training set.

3.3.3 The Effectiveness of SAOA

In this section, we focus on investigating the effectiveness of the proposed SAOA model. We conduct extensive ablation studies to evaluate the contributions of each component and the benefits of different input modalities.

Methods	Verb Top-1	Noun Top-1
Baseline	54.6	23.8
SA w/o CSG	57.0	32.6
SA w/o Gating	57.2	33.6
SA w/o Cross-Stream	57.4	33.2
SA w/o ARM	56.6	32.7
SA	**57.7**	**34.8**

Table 3.1 : The effectiveness of Symbiotic Attention (SA) for **verb prediction** and **noun prediction** on the EPIC-Kitchens validation set. "ARM" denotes the Action-attended Relation Module. "CSG" denotes the Cross-Stream Gating.

The effectiveness of the symbiotic attention

Ablation studies of SA. The symbiotic attention (SA) consists of two modules, *i.e.*, Cross-Stream Gating (CSG), and Action-attended Relation Module (ARM). We evaluate each component on the Epic-Kitchens validation set for both verb and noun classification. We use R-50 as the backbone and RGB data as the input. The results are shown in Table 3.1.

"*Baseline (noun)*" uses a single branch backbone for noun classification. The cross-stream gating module enables mutual communication between the verb branch and the noun branch and re-calibrates the fused features. We implement "*SA w/o CSG*" by performing ARM with the single stream. Specifically, we utilize the global noun feature to attend the object-centric matrix produced by the global alignment module. "*SA w/o CSG*" obtained 32.6% top-1 accuracy, which is 2.2% worse than the unified symbiotic attention. The performance comparison between symbiotic attention and "*SA w/o CSG*" validates the effectiveness of the CSG module. Furthermore, we decompose CSG into two parts, *i.e.*, Cross-Stream and Gating. We

aim to investigate the impact of each component. "*SA w/o Cross-Stream*" indicates using the same stream to gate and attend the noun features. "*SA w/o Gating*" indicates utilizing the feature from the verb stream to attend the object-centric matrix. Specifically, without the cross-stream operation, the performance drops from 34.8% to 33.2%, which confirms the importance of the interaction between the two branches. Without the gating operation, the performance drops from 34.8% to 33.6%, which shows the benefits of our gating mechanism in feature normalization.

We now study the effectiveness of the ARM module. ARM can select the most action-relevant information from the object-centric features and explore the relationships in the spatio-temporal context. The performance drops from 34.8% to 32.7% when ARM is not used, which demonstrates the effectiveness of ARM in action-relevant information selection.

For verb classification, the unified SA outperforms the baseline model by 3.1%. Without the Cross-Stream Gating (CSG), the performance drops by 0.7%. This demonstrates the effectiveness of CSG for verb classification. Specifically, without the gating operation, the performance drops from 57.7% to 57.2%. The performance drops by 0.3% without the cross-stream operation. Moreover, when ARM is not used, the performance drops from 57.7% to 56.6%, which shows the benefits of the action-attended reasoning for verb classification.

SA outperforms other aggregation operations. We first study the effectiveness of our symbiotic attention only using the object detection feature. We directly apply average pooling and max pooling on the object detection features for noun classification. We denote the two pooling methods as "*Det Feat+Avg Pooling*" and "*Det Feat+Max Pooling*", respectively. The results are shown in Table 3.2. "*SA (Det Feat only)*" performs symbiotic attention on object features without the integration of the global noun feature. "*SA (Det Feat only)*" achieves 30.4% on top-1 accuracy,

Methods	Top-1 Accuracy
Det Feat+Avg Pooling	24.5
Det Feat+Max Pooling	25.6
SA (Det Feat only)	30.4
Noun + Det Feat	31.2
SA + Local Alignment	33.6
SA + Global Alignment	**34.8**

Table 3.2 : Comparisons between our symbiotic attention and other aggregation methods for **noun prediction** on the EPIC-Kitchens validation set. "Noun" denotes the global feature from NounNet. "Det Feat" is the location-aware object features.

which outperforms the average pooling baseline and the max-pooling baseline by 5.9% and 4.8%, respectively. The result confirms the superiority of our attention mechanism.

"*Noun+Det Feat*" is one of the baselines to integrate noun features and object detection features, which utilizes the concatenated global noun feature and the max-pooled object feature for classification. "*Noun+Det Feat*" introduces the location-aware object information and uses a simple fusion method to incorporate the location-aware object information. Our symbiotic attention outperforms "*Noun+Det Feat*" by 3.6% on top-1 accuracy (34.8% v.s 31.2%), which demonstrates our symbiotic attention is more effective than the simple aggregation method.

The effectiveness of the global alignment for noun classification

We first conduct the experiment of performing local alignment for the noun branch. The results are shown in the last two rows in Table 3.2. Compared to the

Methods	Verb	Noun	Det Feat	Det Box	R-50	I3D
Baseline (RGB)	✓	-	-	-	54.6	53.2
Verb+Noun Fusion (RGB)	✓	✓	-	-	54.7	53.7
SAP (RGB)	✓	✓	✓	-	55.9	54.3
SAOA (RGB)	✓	✓	✓	✓	**57.7**	**55.1**

Table 3.3 : Ablation study for **verb prediction** using **RGB** data as inputs. We evaluate the comparisons two backbones, *i.e.*, R-50 and I3D. The top-1 results are reported on the EPIC-Kitchens validation set. "Det Feat" denotes the object detection feature. "Det Box" denotes the location of the object detection proposal.

model using global alignment for noun classification, the model with local alignment on the noun feature is 1.2% lower in top-1 accuracy on the EPIC-Kitchens validation set. The results show that the global alignment is more proper than local alignment for the noun classification. As the noun features and the local detection features are mutually complementary with less semantic gap, we use global alignment for the integration of the noun feature and the object detection feature.

The effectiveness of the local alignment for verb classification.

In our SAOA framework, the location-aware object information is globally aligned with the noun feature for noun classification. The location-aware information is locally aligned with the verb feature for verb classification. In this section, we quantitatively compare our SAOA with the previous work SAP [98] and demonstrate that the local alignment approach is better than the global alignment for *verb classification*. We use **RGB** data as inputs, and evaluate the performance on both R-50 and I3D. The top-1 results are shown in Table 3.3.

"*Baseline*" denotes the single branch verb classification model. In "*Verb+Noun*

Fusion", we train a verb classifier with the concatenation of the global noun feature and the verb feature. *SAP* [98] utilizes the global alignment for verb classification. We observe that "*Verb+Noun Fusion*" slightly improves the "*Baseline*" classification model. It shows a simple fusion method does not help to improve verb classification. Compared to "*Baseline*", *SAP* obtain a 1.3% improvement on R-50 and a 1.1% improvement on I3D. This clearly shows that *SAP* well integrates all three sources of information. Compared to the combination of the *global* motion feature and object features in *SAP*, our SAOA leverage a *local alignment* method that can alleviate the semantic gap between detection features and motion features. Our SAOA R-50 model outperforms the SAP R-50 model by 1.8% on top-1 accuracy. Our SAOA I3D model also consistently outperforms the SAP I3D model by 0.8% on top-1 accuracy. Our SAOA significantly outperforms the "*Baseline*" model. We obtained 3.1% and 1.9% improvements for R-50 and I3D, respectively. Compared to the SAP R-50 results on the test sets in Table 3.6, our SAOA R-50 boosts the verb top-1 accuracy from 63.2% to 64.0% and from 53.2% to 55.1% on the test seen set and unseen set, respectively. This demonstrates that our local alignment is effective for verb classification.

Benefit of the multi-modal fusion

Inspired by the two-stream network [85, 106], we aim to leverage a late fusion of the predictions from the SAOA RGB model and the SAOA Flow model to further boost the performance. We conduct the experiments using I3D as the backbone, and we report the results for both verb classification and noun classification on the EPIC-Kitchens validation set in Table 3.4. Our SAOA based on the I3D backbone with the "RGB+Flow+Obj" inputs achieves the highest performance. "SAOA (RGB+Flow+Obj)" outperforms "SAOA (RGB+Obj)" and "SAOA (Flow+Obj)" by 3.5% and 4.7% on top-1 accuracy for verb classification, respectively. For the noun

Methods	Verb Top-1	Noun Top-1
Our SAOA (RGB+Obj)	55.1	34.7
Our SAOA (Flow+Obj)	56.9	35.0
Our SAOA (RGB+Flow+Obj)	**60.4**	**37.4**

Table 3.4 : Two-stream SAOA for both verb classification and noun classification.

classification scenario, we observe a 2.7% and a 1.6% improvements when comparing "SAOA (RGB+Flow+Obj)" with "SAOA (RGB+Obj)" and "SAOA (Flow+Obj)", respectively. This shows that our two-stream SAOA framework is capable of integrating benefits from both RGB and Flow inputs.

3.3.4 Comparison with State-of-the-art Results

We compare our model with the following state-of-the-art methods. ***TSN*** [78] is a 2D CNN model for video recognition. The performance is provided by the dataset authors. ***ORN*** [3] introduces object relation reasoning upon detection features, while the interactions between the verb and noun branches are largely ignored. ***R(2+1)D 34*** [28] indicates the CNN model pre-trained at a very large scale dataset IG-Kinetics (over 65 million videos). ***LFB*** [103] combines Long-Term Feature Banks (detection features) with 3D CNN to improve the accuracy of object recognition. "LFB Max" denotes their best operation on EPIC-Kitchens, which leverages max pooling for feature bank aggregation. ***LSTA*** [89] is an attention-based method, they only report the top-1 action accuracy on the test set. ***TBN*** [50] takes the RGB, Flow, and Audio modalities as input and performs mid-level fusion instead of late fusion. ***SAP*** [98] is our previous work which utilizes global alignment to integrate object features for both verb and noun branches. It also benefits from the symbiotic attention mechanism, and is trained on the R-50 backbone with the

Method	Input Type	Pre-training	Actions		Verbs		Nouns	
			top-1	top-5	top-1	top-5	top-1	top-5
ORN [3]	RGB+Obj	ImageNet	-	-	40.9	-	-	-
R(2+1)D-34 [28]	RGB	IG-Kinetics	22.5	39.2	56.6	**83.5**	32.7	55.5
LFB Max [103]	RGB+Obj	Kinetics+ImageNet	22.8	41.1	52.6	81.2	31.8	56.8
SAP (R-50) [98]	RGB+Obj	Kinetics	25.0	44.7	55.9	81.9	35.0	60.4
Baseline (R-50)	RGB	Kinetics	19.5	36.0	54.6	80.9	23.8	45.1
SAOA (R-50)	RGB+Obj	Kinetics	25.7 (6.2↑)	45.9	57.7	82.3	34.8	59.7
Baseline (R-50)	Flow	Kinetics	16.6	32.8	53.2	79.6	19.7	40.7
SAOA (R-50)	Flow+Obj	Kinetics	24.7 (8.1↑)	43.0	56.1	81.3	33.6	58.7
Baseline (R-50)	RGB+Flow	Kinetics	22.0	40.2	59.3	83.3	27.7	50.9
Our SAOA (R-50)	RGB+Flow+Obj	Kinetics	27.9 (5.9↑)	47.5	**61.0**	**83.8**	36.1	61.6
Baseline (I3D)	RGB	Kinetics+ImageNet	20.5	39.2	53.2	80.4	26.2	51.3
Our SAOA (I3D)	RGB+Obj	Kinetics+ImageNet	24.3 (3.8↑)	44.3	55.1	80.1	34.7	61.4
Baseline (I3D)	Flow	Kinetics+ImageNet	17.9	35.6	54.5	79.9	22.7	45.6
Our SAOA (I3D)	FLow+Obj	Kinetics+ImageNet	25.2 (7.3↑)	43.1	56.9	79.7	35.0	59.7
Baseline (I3D)	RGB+Flow	Kinetics+ImageNet	23.3	43.1	59.7	83.2	29.9	56.0
Our SAOA (I3D)	RGB+Flow+Obj	Kinetics+ImageNet	**28.8** (5.5↑)	**48.4**	60.4	82.8	**37.4**	**63.8**

Table 3.5 : The comparison with the baseline models and state-of-the-art methods on the EPIC-Kitchens validation set. "Obj" indicates the method leverages the information from the object detection model. ↑ indicates the improvement of our method compared to the baseline.

RGB input modality.

Table 3.5 and Table 3.6 summarize the top-1 and top-5 accuracy for action, verb and noun predictions on the EPIC-Kitchens dataset. We develop our approach with R-50 and I3D backbone, and we leverage both RGB and optical flow as the input types for I3D and only RGB frames for R-50. In the Pre-training column, "Kinetics" indicates the backbone is pre-trained on Kinetics[6] directly. "Kinetics+ImageNet" indicates the backbone is pre-trained using the I3D [6] strategy, which first initializes the 3D CNN with the inflated weights of the 2D CNN pre-trained on ImageNet [15]

Method	Input Type	Pre-training	Actions		Verbs		Nouns	
			top-1	top-5	top-1	top-5	top-1	top-5
Test seen								
TSN Fusion [78]	RGB+Flow	ImageNet	25.4	45.7	54.7	87.2	40.1	65.8
R(2+1)D-34 [28]	RGB	IG-Kinetics	34.4	54.2	63.3	87.5	46.3	69.6
LSTA [89]	RGB+Flow	ImageNet	30.2	-	-	-	-	-
LFB Max [103]	RGB+Obj	Kinetics+ImageNet	32.7	55.3	60.0	88.4	45.0	71.8
TBN [50]	RGB+Flow+Audio	Kinetics+ImageNet	34.8	56.7	64.8	**90.7**	46.0	71.3
SAP R-50 [98]	RGB+Obj	Kinetics	34.8	55.9	63.2	86.1	48.3	71.5
Our SAOA (R-50)	RGB+Obj	Kinetics	37.0	58.3	64.0	88.0	**49.6**	**73.2**
Our SAOA (I3D)	RGB+Obj	Kinetics+ImageNet	33.8	55.3	63.6	87.4	46.1	70.0
Our SAOA (I3D)	Flow+Obj	Kinetics+ImageNet	33.4	54.7	63.8	86.8	45.7	69.2
Our SAOA (I3D)	RGB+Flow+Obj	Kinetics+ImageNet	**37.7**	**59.2**	**67.6**	89.2	47.8	71.8
Test Unseen								
TSN Fusion [78]	RGB+Flow	ImageNet	14.8	29.8	46.1	76.7	24.3	49.3
R(2+1)D-34 [28]	RGB	IG-Kinetics	23.7	39.1	55.5	80.9	33.6	56.7
LSTA [89]	RGB+Flow	ImageNet	15.9	-	-	-	-	-
LFB Max [103]	RGB+Obj	Kinetics+ImageNet	21.2	39.4	50.9	77.6	31.5	57.8
TBN [50]	RGB+Flow+Audio	Kinetics+ImageNet	19.1	36.5	52.7	79.9	27.9	53.8
SAP R-50 [98]	RGB+Obj	Kinetics	23.9	40.5	53.2	78.2	33.0	58.0
Our SAOA (R-50)	RGB+Obj	Kinetics	23.3	41.2	55.1	79.9	32.3	57.1
Our SAOA (I3D)	RGB+Obj	Kinetics+ImageNet	21.9	42.1	52.9	79.9	31.7	58.5
Our SAOA (I3D)	Flow+Obj	Kinetics+ImageNet	23.2	42.4	55.5	80.1	32.6	58.1
Our SAOA (I3D)	RGB+Flow+Obj	Kinetics+ImageNet	**25.8**	**45.1**	**58.1**	82.6	**34.4**	**60.4**

Table 3.6 : The comparison with the baseline models and state-of-the-art methods on the EPIC-Kitchens test set. "Obj" indicates the method leverages the information from the object detection model.

and then trains the 3D CNN on Kinetics. "IG-Kinetics" indicates the backbone is pre-trained on a large-scale dataset, *i.e.*, IG-Kinetics [28], with weak supervision.

Table 3.5 shows the results of our method and the baselines on the EPIC-Kitchens validation set. The proposed SAOA outperforms the baselines by a large margin

	Method	Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
		Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Seen	Attention Clusters [68]	40.4	19.4	11.1	78.1	41.7	24.4	21.2	9.7	2.5	14.9	11.5	3.4
	TSN Fusion [12]	48.2	36.7	20.5	84.1	62.3	39.8	47.3	35.4	11.6	22.3	30.5	9.8
	LSTA Ensemble [88]	63.3	44.8	35.5	89.0	69.9	57.2	63.2	42.3	19.8	37.8	41.3	21.2
	TBN Ensemble [50]	66.1	47.9	36.7	91.3	72.8	58.6	60.7	44.9	24.0	46.8	43.9	22.9
	Sudhakaran (3rd place)	68.7	49.4	40.0	91.0	72.5	60.2	60.6	45.5	21.8	47.2	45.8	24.3
	action banks (2nd place)	66.7	49.6	41.6	90.1	77.0	64.1	59.4	45.6	25.4	41.7	46.3	27.0
	two-stream SAOA I3D	67.6	47.8	37.7	89.2	71.8	59.3	57.8	42.1	19.6	42.7	44.8	20.7
	Our SAOA (1st place)	**70.4**	**52.9**	**42.6**	90.8	76.6	63.6	60.4	**47.1**	24.9	45.8	**50.0**	26.9
Unseen	Attention Clusters [68]	32.4	12.0	5.6	69.9	31.8	15.7	17.2	3.9	1.8	11.6	7.9	2.6
	TSN Fusion [12]	39.4	22.7	10.9	74.3	45.7	25.3	22.5	15.3	6.2	13.1	17.5	6.5
	LSTA Ensemble [88]	49.4	27.1	20.3	77.5	52.0	37.6	31.1	21.1	9.2	18.7	21.9	14.2
	TBN Ensemble [50]	54.5	30.4	21.0	81.2	55.7	39.4	32.6	21.7	11.0	27.6	25.6	13.3
	action banks (3rd place)	54.6	33.5	27.0	80.4	61.0	46.4	33.6	30.5	15.0	25.3	28.4	18.0
	aptx4869lm (2nd place)	60.1	38.1	27.4	82.0	63.8	45.2	33.6	31.9	16.5	29.3	33.9	20.1
	two-stream SAOA I3D	58.1	34.4	25.8	82.6	60.4	45.1	38.9	28.7	14.8	28.7	30.1	17.5
	Our SAOA (1st place)	**60.4**	37.3	**28.0**	**83.1**	63.7	**46.8**	35.2	**32.6**	**17.4**	29.0	32.8	19.8

Table 3.7 : Comparison with the methods on the leaderboard of EPIC-Kitchens Action Recognition Challenge. The results of Attention Clusters are borrowed from [50].

on all modalities with both two backbones. Specifically, with RGB frames as input, our SAOA R-50 significantly boosts the top-1 accuracy from 19.5% to 25.7% on action classification. With optical flow as input, our SAOA R-50 outperforms the baseline by 8.1% in top-1 accuracy on action recognition, demonstrating that SAOA can incorporate optical flow input effectively. Compared to I3D baselines, our SAOA I3D consistently improves the action recognition performance with different modalities. The remarkable performance gains mainly benefit from the symbiotic attention mechanism and the integration of the location-aware object information. Our SAOA R-50 achieves higher top-1 accuracy than the original SAP on verb prediction and action prediction. The improvement is owing to the integration of the location-aware alignment for the verb feature. Compared to the model with single

modality input, the two-stream SAOA achieves higher accuracy, which demonstrates the effectiveness of the proposed multi-modal fusion strategy.

Our model outperforms the state-of-the-art methods by a large margin on all three evaluation splits, *i.e.*, the validation set, the test seen set and the test unseen set. On the validation set, compared to "LFB Max", which also utilizes the detection features, our two-stream SAOA (I3D) on the action prediction significantly improves the top-1 accuracy from 22.8% to 28.8%. With the same type of input (RGB+Obj), our SAOA (R-50) still outperforms them by 3.0%. The significant improvement mainly benefits from the interactions between the verb branch, noun branch, and the location-aware alignment with the location-aware object information. Although R(2+1)D 34 [28] uses much more videos to train the model, our best model still outperforms them by 6.3% in top-1 accuracy for action classification.

On the test seen set and the test unseen set, compared to the previous state-of-the-art method TBN, our two-stream SAOA (I3D) outperforms the recognition accuracy by a large margin. Specifically, the improvement of top-1 accuracy on the unseen set is 6.7%, 5.4%, and 6.5% for action, verb, and noun, respectively. Compared to our vanilla SAP model [98], our two-stream SAOA (I3D) achieves higher accuracy on all metrics. This demonstrates the effectiveness of the proposed location-aware alignment and the multi-modal fusion strategy.

3.3.5 EPIC-Kitchens Action Recognition Challenge 2020

We further verified the effectiveness of our framework on the EPIC-Kitchens Action Recognition Challenge. Our method achieved first place on both the seen set and unseen set. As shown in Table 3.7, we compare our approach with the top-3 submissions of Action Recognition Challenge and three published works on the leaderboard.

Notably, on the unseen test set, our single model (two-stream SAOA (I3D))

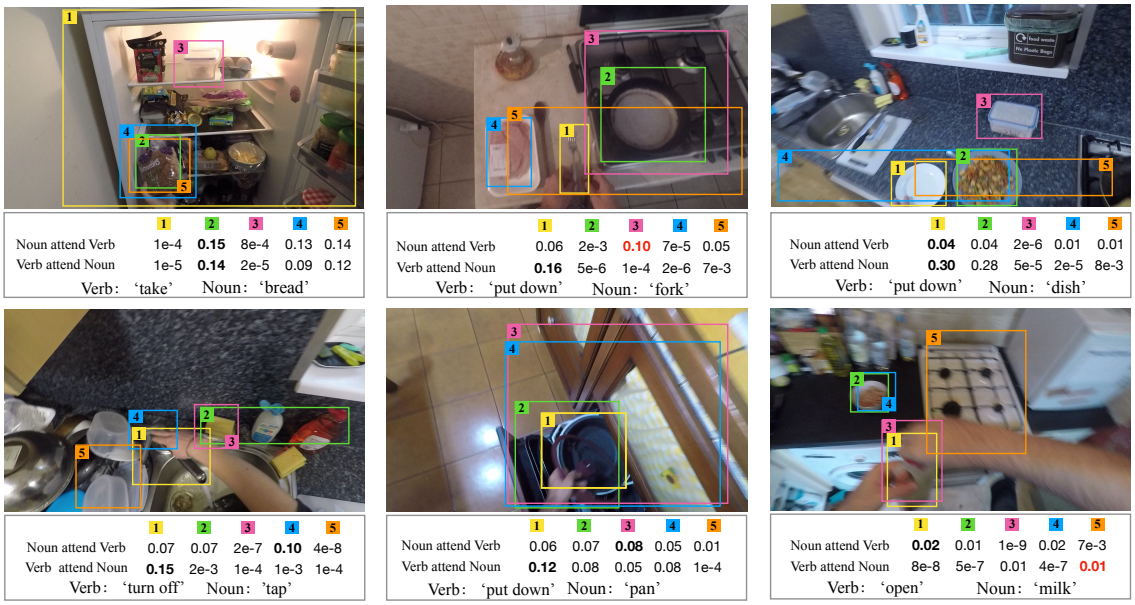

Figure 3.4 : Qualitative results of our SAOA I3D (Flow) model. The colored boxes show the top-5 detected regions and the numbers are the corresponding attention weights generated by our action-attended relation module. Red indicates the failure case.

achieves higher performance than the TBN [50] Ensemble on all evaluation metrics. We also report the result of our final submission to the challenge, which fuses the predictions of six models trained with different backbones and input modalities. For the final action recognition, our ensemble achieves 42.6% top-1 accuracy on seen set and 28.0 % on unseen set, which are higher than the TBN ensemble by 5.9% and 7.0%, respectively. Our result outperforms the second place submission on seen set by 1.0% and 0.6% on unseen set. The final rank is based on top-1 action accuracy.

3.3.6 Visualization

In Fig. 3.4, we show some qualitative results on the EPIC-Kitchens validation set.

The colored boxes in the figure indicate the top confident object proposals gen-

erated by the pre-trained detection model. We do not use labels of detected objects since they are not accurate. Instead, we use the object feature and location to guide the mutual communication of the verb and noun branch. The numbers below each image are the values of ARM attention weights for the five object-centric features.

As illustrated in the first video frame (the left-top one), the ground truth of this video clip is "take bread". Our ARM module generates the highest value for the feature corresponding to the second box where the interaction happened. The weights for the fourth and fifth box is similar to the second box since their locations are very close and also the boxes also contain the object "bread". The distracting objects with the first and third boxes obtain the lowest scores. For the last figure in the second row, our ARM failed to produce correct values for the boxes in the noun branch. This is owing to the large camera motion and occlusion of the objects. According to the qualitative analysis of the six examples, we can observe that the attention weights of the noun branch ("Verb attend Noun") are more accurate than the values of the verb branch ("Noun attend Verb").

3.4 Summary

In this Chapter, we propose a novel framework named Symbiotic Attention with Object-centric feature Alignment (SAOA) for egocentric action recognition. We introduce a local and global alignment method to integrate the location-aware object information. Local motion features are produced to bridge the semantic gap between the motion feature and the object detection feature. We introduce a new attention mechanism called symbiotic attention that interactively leverages sources from the verb branch, the noun branch, and the location-aware object information. We evaluate SAOA on two backbones, two modalities, and the largest egocentric action recognition dataset. Our experimental results demonstrate the effectiveness of our framework, and we significantly outperform the state-of-the-art methods on

the largest egocentric video dataset. In the future, we will explore to suppress background distractors in the convolutional backbones. It is promising to leverage other attention mechanisms to integrate multiple sources of information.

Chapter 4

Recognizing Active Objects in Egocentric Videos via Interactive Prototype Learning

In the previous chapter, I introduce a collaborative modeling framework between verb branch and noun branch with the guidance of local object information. However, running object detectors on high-resolution video frames is computationally expensive, and human annotations are not always available. In this chapter, we develop an egocentric action recognition without the integration of an object detector.

4.1 Introduction

In egocentric videos, *noun* classification is particularly difficult as the active object [23, 25] involved in the action can be surrounded by a considerable number of distracting objects, *e.g.*, the bowl and pan surrounding the active object "potato" in Fig. 4.1. Indeed, *noun* classification tends to have much lower accuracy in egocentric video datatsets [12, 98], and is the bottleneck of the whole action recognition system. Previous methods either use off-the-shelf object detectors [98, 103] or human gaze provided by the datasets [59] as additional cues to improve *noun* recognition.

In this Chapter, we propose to improve active object recognition in egocentric videos by leveraging the information learned from actor motion understanding. The active objects often locate on the area where the motion is performing. Moreover, actor motion carries the intention of the actor and is often the dominant signal in the egocentric video, which can serve as a reliable source to improve active object recognition.

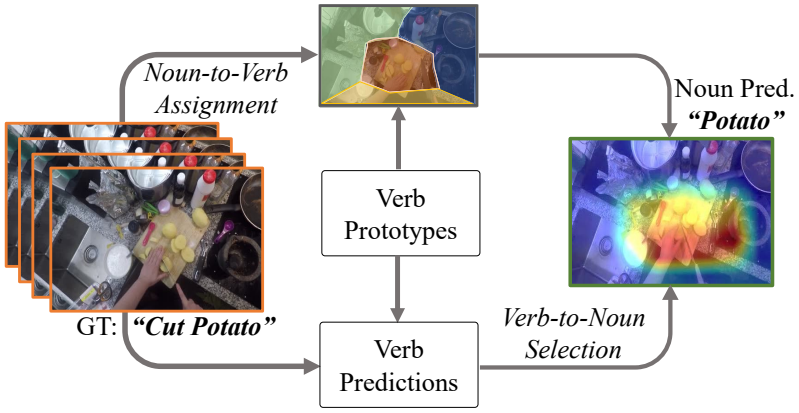

Figure 4.1 : The motivation of **I**nteractive **P**rototype **L**earning (IPL) framework is to collaboratively learn judicious location-aware spatio-temporal features for more accurate *noun* (active object) classification.

More precisely, we devise an end-to-end **I**nteractive **P**rototype **L**earning (IPL) framework to disentangle active object features from distracting object features (Fig. 4.1). IPL consists of two interactive operations, *i.e.*, *noun*-to-*verb* assignment and *verb*-to-*noun* selection. The two operations collaboratively extract judicious location-aware spatio-temporal features for *noun* classification. We demonstrate that the extracted features are more relevant to target action candidates.

In *noun*-to-*verb* assignment, we introduce a set of *verb prototypes* to represent the primary actor intentions. Each prototype corresponds to a category in the *verb* vocabulary. These trainable prototypes offer a distinctive supervision signal for each actor motion. After obtaining spatio-temporal features from the last convolutional layer of a 3D CNN [94, 6], we aim to aggregate spatio-temporal features based on their similarities to the *verb prototypes*. To this end, each verb prototype obtains an aggregated feature that is most relevant to it. We introduce an assignment and a grouping operation to obtain these action-centric representations, in a similar spirit as NetVLAD [46, 2]. In addition, we should enforce the semantic alignment between each prototype and its corresponding label in the *verb* vocabulary. We

propose to directly optimize *verb prototypes* by regarding them as the weights in a cosine classifier. In this way, the prototypes can be jointly optimized during *verb* classification, utilizing the ground-truth verb labels as supervision. This simple operation effectively maintains the semantics of *verb* prototypes.

In *verb*-to-*noun* selection, we aim to select the most action-related features for the final *noun* classification. We propose to filter the set of aggregated features obtained from the previous stage by the guidance from *verb* classification predictions. Motivated by the empirical observation that *verb* classification is significantly better than *noun* classification [12, 13], we use top-K predictions from verb classification to select the corresponding aggregated features and remove the remaining "irrelevant" features. The *verb*-to-*noun* selection stage generates more discriminative features for the final noun classification.

With extensive experiments and detailed ablation studies, we demonstrate that IPL outperforms the state of the art on three large-scale egocentric video dataset (*i.e.*, EPIC-KITCHENS-100 [13], EPIC-KITCHENS-55 [12] and EGTEA [59]), despite being simple and easy to train with different video backbones [6, 94].

4.2 Interactive Prototype Learning

4.2.1 Overview

Given an input video clip $\boldsymbol{X}$, the goal is to classify it to M verb classes and N noun classes. The underlying action can be inferred from the verb and noun prediction results. As shown in Figure 4.2, we first extract the spatio-temporal feature map $\boldsymbol{\phi}_\theta(\boldsymbol{X}) \in \mathbb{R}^{T\times H\times W\times C}$ from the last convolutional layer of the 3D CNN backbone $\boldsymbol{\phi}_\theta$, where θ is the parameters of the CNN, T is the temporal length, C is the number of channels and $H \times W$ is the spatial resolution.

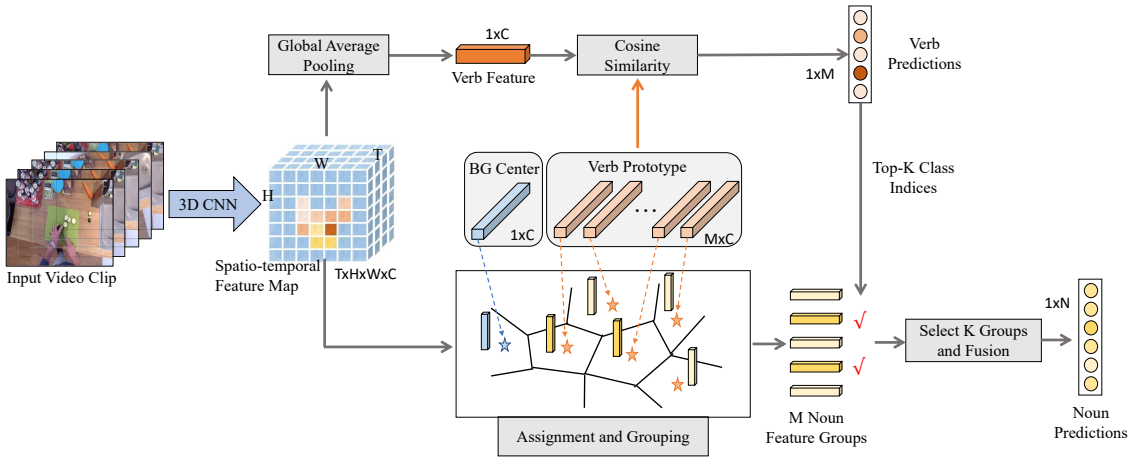

Figure 4.2 : Our Interactive Prototype Learning (IPL) framework. The feature map of size $T \times H \times W \times C$ is extracted from the last convolutional layer of the 3D CNN backbone. To facilitate the interaction between the verb branch and the noun branch, we introduce a set of *verb prototypes* shared across the two branches. A background prototype is introduced to filter the action-irrelevant information from the spatio-temporal feature map. Each prototype is a C-dimensional vector and is random initialized during training. Verb prediction is obtained by computing the cosine similarity between the average pooled verb feature and the verb prototypes. For noun prediction, the feature map is decomposed and grouped by soft-assigning each feature to the prototypes. We select the most relevant K groups based on verb predictions to generate the final noun representation. The 3D CNN backbone and IPL are jointly trained in an end-to-end manner.

The core idea of IPL is to utilize verb features to guide the learning of action-centric object features. Specifically, we introduce M *verb prototypes* $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M\} \in \mathbb{R}^{M \times C}$, where each prototype corresponds to a verb class and represent one type of motion pattern from the actor. All M prototypes are shared between the verb and noun branches in order to enable interactive learning.

In the verb branch, we obtain the C-dimensional verb feature vector by applying

global average pooling on $\boldsymbol{\phi}_\theta(\boldsymbol{X})$ (Section 4.2.2). Unlike the conventional linear classifier implemented by a fully connected layer, we use a simple nearest neighbor classifier with a cosine similarity [29] between the verb feature and M verb prototypes. This simple strategy allows us to directly apply the strong supervision from the verb ground-truth to learn more semantic verb prototypes.

In noun classification, we design two interactive operators to extract location-aware features from $\boldsymbol{\phi}_\theta(\boldsymbol{X})$ to perform the noun classification. In *noun*-to-*verb* assignment operator (Section 4.2.3), we decompose $\boldsymbol{\phi}_\theta(\boldsymbol{X})$ into THW C-dimensional features, and assign each feature to M verb prototypes and one additional background prototype to catch irrelevant background information. This converts the THW features into $M+1$ feature groups. In *verb*-to-*noun* selection operator (Section 4.2.3), we select K feature groups corresponding to the top-K verb classes with the highest classification score from the verb branch. The selected K features are then aggregated to obtain the final representation for noun classification.

4.2.2 Verb Classification

Verb Prototype. In egocentric videos, motion is the dominant information for action recognition and indicates the intention of the actor and which object the actor wants to interact with [12, 13]. This motivates us to leverage the actor motion information to improve active object recognition, which is an arguably harder task. Specifically, we propose to learn a prototype for each verb class. We denote the *verb prototypes* as $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_M\}$, where $\boldsymbol{P} \in \mathbb{R}^{M \times C}$. These verb prototypes are intermediate representations to facilitate interaction between the verb and noun classification. They are anchors for grouping spatio-temporal features based on their similarities.

Cosine Classifier. Inspired by recent works [9, 29], We design a simple and effective classifier: Nearest-Neighbors (NN) with ℓ_2-normalized features, usually named cosine classifier. Given the spatio-temporal feature map $\boldsymbol{\phi}_\theta(\boldsymbol{X})$, the verb feature is generated with global average pooling (GAP):

$$\boldsymbol{v} = GAP(\boldsymbol{\phi}_\theta(\boldsymbol{X})), \tag{4.1}$$

where $\boldsymbol{v} \in \mathbb{R}^{1\times C}$. After that, we calculate the cosine similarity between the verb feature and each verb prototype. The verb classification probability q_i for the i-th class is generated after a softmax activation function. Formally,

$$q_i = \frac{exp(\bar{\boldsymbol{v}}\bar{\boldsymbol{p}}_i{}^\top/\tau)}{\sum_{j=1}^{M} exp(\bar{\boldsymbol{v}}\bar{\boldsymbol{p}}_j{}^\top/\tau)}, \tag{4.2}$$

where $\bar{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$ and $\bar{\boldsymbol{p}}_i = \frac{\boldsymbol{p}_i}{\|\boldsymbol{p}_i\|}$ are the l_2-normalized vectors. Here we use a temperature τ to re-scale the similarities following [29, 9]. The temperature τ can help training similarity-based classifier and reduce intra-class variations [9], which is beneficial for learning discriminative video representations.

4.2.3 Noun Classification

Feature Assignment and Grouping

In the egocentric video, the motion from the actor gives strong indications about what object the actor interacts with. This inspires us to leverage the motion features to identify the features from the active objects and suppress the features from distracting objects. We design new operators that can decompose and regroup object features based on their relevance to the actor motion and learn more discriminative features for active object classification.

Feature Assignment. We propose to assign the THW C-dimensional features to the learned prototypes. Besides the aforementioned M verb prototypes, we also introduce a background prototype $\boldsymbol{b} \in \mathbb{R}^{1\times C}$ to catch all the irrelevant features that

do not match to any of the motion patterns from the M verb classes. In total we have $M+1$ prototypes, noted as $\boldsymbol{C} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M, \boldsymbol{b}\}$, where $\boldsymbol{C} \in \mathbb{R}^{(M+1)\times C}$. By assigning THW features to $M+1$ prototypes, we can disentangle the features of the active object from the features of the distracting objects, and select relevant features for noun classification.

Inspired by the success of dot-product attention mechanism [95], we use a simple dot-product operator between the feature vector and the learned prototypes to measure their similarity. A softmax function is applied to the dot product to achieve the soft assignment of the feature to the $M+1$ prototype. For convenience, we reshape the spatio-temporal feature map $\boldsymbol{\phi}_\theta(\boldsymbol{X}) \in \mathbb{R}^{T\times H\times W\times C}$ to a 2D tensor $\boldsymbol{Z} \in \mathbb{R}^{B\times C}$, where $B = T \times H \times W$. For a feature vector $\boldsymbol{z}_i$ from $\boldsymbol{Z}$, the assignment to the prototype $\boldsymbol{c}_j$ is defined as,

$$a_{i,j} = \frac{exp(\boldsymbol{z}_i \boldsymbol{c}_j^\mathsf{T})}{\sum_{k=1}^{M+1} exp(\boldsymbol{z}_i \boldsymbol{c}_k^\mathsf{T})}, \tag{4.3}$$

where $a_{i,j}$ is an element from the soft assignment matrix $\boldsymbol{A'} \in \mathbb{R}^{B\times(M+1)}$. To simplify, we remove the assignments belong to the background prototype $\boldsymbol{b}$ from $\boldsymbol{A'}$ as they are considered irrelevant for active object recognition. We end up with a new assignment matrix $\boldsymbol{A} \in \mathbb{R}^{B\times M}$.

Feature Grouping. We aggregate all the assigned features on each prototype to obtain M feature groups. The aggregation operation can be performed by a matrix multiplication as follows,

$$\boldsymbol{G} = \boldsymbol{A}^\mathsf{T} \boldsymbol{Z}, \tag{4.4}$$

where $\boldsymbol{G} \in \mathbb{R}^{M\times C}$ denotes the feature groups on the M prototypes. $\boldsymbol{g_i} \in \mathbb{R}^C$ is the i-th row of $\boldsymbol{G}$, and represents the aggregated feature belongs to prototype $\boldsymbol{p}_i$. $\boldsymbol{g_i}$ includes all the information from both the actor motion and the active object. To

obtain the feature of the active object, we compute a residual between the grouped feature $\boldsymbol{g_i}$ and the verb prototype:

$$\boldsymbol{g}_i^{noun} = \boldsymbol{g_i} - \sum_{k=1}^{B} a_{k,i}\boldsymbol{p}_i, \tag{4.5}$$

where $a_{k,i}$ is the element in $\boldsymbol{A} \in \mathbb{R}^{B\times M}$. The normalization on $\boldsymbol{p}_i$ is for calibration so that it is on the same scale as $\boldsymbol{g}_i$. $\boldsymbol{g}_i^{noun}$ is the final noun feature w.r.tprototype $\boldsymbol{p}_i$.

Group Selection and Noun Classification

After feature assignment and grouping, we obtain a set of features $\boldsymbol{G}^{noun} = \{\boldsymbol{g}_1^{noun}, \boldsymbol{g}_2^{noun}, \ldots, \boldsymbol{g}_M^{noun}\}$ corresponding to M verb prototypes. Given a trimmed video clip, we want to identify the features that are most related to the motion of the actor, and suppress the features that may come from irrelevant background or distracting objects.

Towards this end, we propose to simply select top-K features from $\{\boldsymbol{g}_1^{noun}, \boldsymbol{g}_2^{noun}, \ldots, \boldsymbol{g}_M^{noun}\}$ based on their verb classification scores. We sort M verb predictions in decreasing order. We denote the indices of the top-K classes with the highest scores as $\{i_1, i_2, \ldots, i_K\}$. Then the top-$K$ selected features are $\{\boldsymbol{g}_{i_1}^{noun}, \boldsymbol{g}_{i_2}^{noun}, \ldots, \boldsymbol{g}_{i_K}^{noun}\}$.

We apply l_2-normalization to each selected feature and concatenate them to generate the feature $\boldsymbol{n'} \in \mathbb{R}^{K\times C}$. We then use a layer $f_{\boldsymbol{w}}$ parameterized by $\boldsymbol{w}$ to enhance feature $\boldsymbol{n'}$, which also reduces its dimension to C. We obtain the final noun representation $\boldsymbol{n} = f_{\boldsymbol{w}}(\boldsymbol{n'})$, which can be directly used for classification. Similar to verb classification, we simply use a cosine classifier for noun classification to reduce intra-class variations. In our implementation, we instantiate $f_{\boldsymbol{w}}$ with an 1D convolutional layer with batch normalization [45] followed by ReLU activation. Note that the number of additional parameters introduced by $f_{\boldsymbol{w}}$ is negligible.

Relations to NetVLAD. The implementation of IPL shares similar components with the NetVLAD layer [2, 30, 73], if we consider the verb prototypes as the NetVLAD clusters. Unlike our verb prototypes, the clusters in NetVLAD are not trained with direct supervision from a classification loss. The semantic meaning of the NetVLAD clusters is unclear, as they only serve as anchors in the feature space for clustering. In contrast, our verb prototypes are directly optimized with a loss for verb classification. Each verb prototype can be considered as a representation to capture the motion patterns of a verb class. Benefit from this design, our learned prototypes can be directly used for Verb classification with a simple nearest neighbor classifier. Note that the verb prototypes are also used to assign features for noun classification and play the role of bridging the two tasks (*i.e.*, verb and noun classification) for egocentric video recognition. Additional supervision from noun classification further enhances the semantic meaning of the learned prototypes. Instead of concatenating the features from all the clusters in NetVLAD, we only select top-K aggregated features as we aim to disentangle the features from the active object and the features from the distracting objects.

4.2.4 Training and Inference

During training, we use cross-entropy loss for classification. The overall training objective is to minimize the sum of the verb classification loss and the noun classification loss. The 3D CNN backbone and the Interactive Prototypical Network are jointly optimized in an end-to-end manner. During inference, given an input video clip, the framework produces verb and noun predictions simultaneously. The action predictions are generated by combining verb and noun predictions.

Split	Method	Overall						Unseen Participants			Tail Classes		
		Top-1 Accuracy			Top-5 Accuracy			Top-1 Accuracy			Top-1 Accuracy		
		Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Val	Chance [13]	10.42	1.70	0.51	38.43	8.14	2.54	10.59	1.88	0.57	1.13	0.31	0.10
	TSN [97]	60.26	46.75	33.42	89.41	73.62	55.51	49.11	37.28	24.04	30.06	21.35	14.36
	TRN [122]	65.05	45.21	34.42	90.14	72.09	56.54	55.40	37.46	26.95	33.86	18.68	13.95
	TBN [50]	65.26	47.49	36.08	90.32	73.94	58.04	58.22	38.31	28.36	**38.24**	**26.20**	**18.89**
	SlowFast [24]	65.92	49.74	38.54	89.98	75.02	58.27	56.81	39.72	28.92	35.57	22.13	16.77
	TSM [62]	68.35	48.99	38.48	**91.04**	74.94	60.51	58.50	39.53	29.39	37.39	23.07	18.41
	I3D* [6]	66.84	48.48	37.58	90.57	73.94	59.13	60.09	39.91	29.77	36.08	17.37	15.27
	IPL I3D	**67.82**	**50.87**	**39.87**	**90.85**	**76.07**	**61.95**	**61.22**	**41.03**	**30.89**	36.76	23.26	18.23
Test	Chance [13]	10.68	1.79	0.55	37.71	8.35	2.69	9.37	1.90	0.59	0.97	0.39	0.12
	TSN [97]	58.43	46.54	32.79	87.27	72.49	53.12	52.04	42.09	26.30	24.76	14.91	10.41
	TRN [122]	62.56	45.70	34.41	88.24	71.37	54.65	57.49	40.85	28.69	27.24	13.42	11.20
	TBN [50]	62.40	46.50	35.11	88.74	72.24	55.17	56.57	41.78	29.46	29.41	18.55	13.47
	SlowFast [24]	63.91	49.01	36.85	88.71	74.79	56.54	56.98	44.62	29.98	30.00	17.81	12.94
	TSM [62]	65.51	48.48	37.58	89.39	73.46	58.04	**59.66**	43.16	30.41	29.76	15.84	13.15
	IPL I3D	**65.66**	**49.74**	**38.43**	**89.66**	**74.98**	**59.28**	59.12	**45.26**	**32.17**	**32.17**	**20.34**	**15.51**

Table 4.1 : The comparison with the baseline models and state-of-the-art methods on the **EPIC-KITCHENS-100** dataset. Note that TSN [122], TRN [122], TSM [62] and our models are two-stream models, while TBN leverages an extra audio stream for classification. "*" indicates our I3D implementation.

4.3 Experiment

4.3.1 Dataset

EPIC-KITCHENS-100 [13] is a recently introduced large-scale egocentric video dataset. It consists of 100-hour videos and contains 89,979 segments of fine-grained actions. Compared to EPIC-KITCHENS-55 [12], the annotations are denser and more accurate. It has 97 verb classes and 300 noun classes. The action of each video segment is defined by the verb-noun pair (*e.g.*, "cut potato", "take knife"). The action segments are splitted into Train/Val/Test by the dataset developers, and

Val/Test splits contain two subsets, *i.e.*, Unseen Participants and Tail Classes. We report top-1 and top-5 accuracy on overall segments and the subsets on Val/Test splits.

EPIC-KITCHENS-55 [12] is the previous version of EPIC-KITCHENS-100 [13]. It is a large-scale dataset in first-person vision. It consists of 55 hours of recordings capturing all daily activities in the kitchens. The performed activities are non-scripted, which makes the dataset very challenging and close to real-world data. The dataset contains 39,594 action segments which are annotated with 125 verb classes and 321 noun classes. We split the original training set to a new training and validation set following [3]. We report the top-1 accuracy of the verb, noun, and action on the validation set.

EGTEA [59] is a large-scale egocentric video dataset which consists of 10321 video clips annotated with 19 verb classes, 51 noun classes, and 106 action classes. We report mean class accuracy on the three different Train/Val splits.

4.3.2 Implementation Details

We train two backbone networks with the proposed Interactive Prototype Learning, *i.e.*, I3D [6] and R(2+1)D-34 [94]. For I3D, we train both spatial and temporal streams with 64 RGB frames or flow images as input. The backbone is initialized using the Kinetics [6] pretrained weights. We adopt the stochastic gradient descent (SGD) with a momentum 0.9 and weight decay 0.0005 to optimize the parameters for 30 epochs. The learning rate is initialized to 0.006 and then changed to 0.0006 in the last 10 epochs. The batch size is set to 32. During training, the spatial size of input clip is 224×224. Random scaling, random cropping and horizontal flipping are deployed as data augmentation. During inference, we resize the frame to the size 256×256 and feed them to the model withour cropping. We then average the predictions of ten uniformly sampled clips as the final video-level predictions. For

R(2+1)D-34 [94], we train the RGB stream using the IG-Kinetics [28] pretrained weights as initialization. The learning rate is set to 0.0004 and drops by ten times every nine epochs. We use SGD with a momentum 0.9 and weight decay 0.0005 to optimize the model for 20 epochs. We use 32 frames as input and the spatial size is 112×112 during training and 128×128 during testing. And the same data augmentation and multi-crop testing strategy are used on the model.

4.3.3 Comparison with State of the Arts

Results on EPIC-KITCHENS-100

We compare our Interactive Prototype Learning framework with the state-of-the-art methods on both the validation set and the test set of EPIC-KITCHENS-100 [13]. TSN [97], TRN [122] and TSM [62] are designed based on 2D CNN. All the three models employ a two-stream approach and take both RGB frames and optical flow as input. TBN [50] utilizes RGB, optical flow and audio modalities as inputs. It performs mid-level fusion instead of late fusion in a two-stream strategy. SlowFast [24] combines two pathway 3D CNNs with different input resolutions and frame rates.

We build our interactive prototype learning framework based on I3D [6]. To fairly compare to TSN [97], TRN [122] and TSM [62], we also train a two-stream I3D with both RGB frame and optical flow as input. Besides, we set a two-stream I3D baseline with two linear classifiers cascaded to the CNN backbone. As shown in Table 4.1, our IPL I3D outperforms the baseline model on all evaluation metrics. On the Val set, our Interactive Prototype Learning (**IPL**) improves the overall top-1 accuracy of noun classification by 2.39%, boosting the accuracy from 48.48% to 50.87%. The performance gain mainly benefits from the *noun*-to-*verb* grouping and *verb*-to-*noun* selection operation. By introducing the interactions with verb prototypes, the most action-relevant features can be leveraged for noun classification.

Method	Act@1	Verb@1	Noun@1
R50-NL [103]	19.0	49.8	26.1
R(2+1)D-34* [28]	22.5	56.6	32.7
SlowFast [109]	21.9	55.8	27.4
I3D	23.5	59.6	31.3
IPL (I3D)	24.5	59.8	33.2 (**+1.9**)
R(2+1)D-34	23.6	60.5	31.1
IPL (R(2+1)D-34)	25.4	60.7	35.5 (**+4.4**)

Table 4.2 : Comparisons of 3D CNN backbones on the EPIC-KITCHENS-55 validation set. *: note that [28] leverages two R(2+1)D-34 backbones. One is for verb classification and the other is for noun classification. Our "IPL (R(2+1)D-34)" and "R(2+1)D-34" share the network backbone.

The verb recognition accuracy is also slightly improved (+0.98%). This is because the interactive learning scheme could also enhance the verb prototype learning. Due to the considerable improvement on noun recognition, the overall action recognition top-1 accuracy is boosted by 2.29% (from 37.58% to 39.87%). We outperform other state-of-the-art baselines (*e.g.*, TSN [97], TRN [122], TSM [62] and SlowFast [24]) on overall top-1 accuracy and the unseen participants set. For instance, we outperform TSM by 1.39% on overall top-1 action accuracy. On the unseen participant subset, we obtain a 1.5% gain for action classification. Though our results on tail classes are lower than TBN on the validation set, we consistently outperform all baselines on the public test set across all subsets. Notably, we obtain 1.36% gain of action classification on tail classes. Our IPL achieves state-of-the-art action recognition performance on overall data. These results demonstrate the effectiveness and generalization capability of the proposed method.

Method	Obj	Act@1	Verb@1	Noun@1
ORN [3]	✓	-	40.9	-
LFB Max [103]	✓	22.8	52.6	31.8
SAP [98]	✓	25.0	55.9	35.0
IPL (R(2+1)D-34)	✗	25.4	60.7	35.5

Table 4.3 : Comparisons with the state-of-the-art methods using object detection annotations on the EPIC-KITCHENS-55 validation set.

Results on EPIC-KITCHENS-55

Backbone comparisons. We compare to state-of-the-art 3D CNN backbones on the EPIC-KITCHENS-55 validation set. We report the baseline of both I3D and R(2+1)D-34 in Table 4.2. On the I3D backbone, our IPL outperforms the I3D baseline by 1.9% for noun classification. Notably, on the R(2+1)D-34 backbone, IPL outperforms the baseline with 4.4% for noun classification. These results clearly show that IPL works effectively on EPIC-KITCHENS-55. With the clear gains in noun classification, the action classification accuracy is also improved on both backbones. For instance, there is a 1.6% gain when we compare IPL (R(2+1)D-34) with its baseline.

Comparisons to methods that utilize object detection annotations. As the EPIC-KITCHENS-55 dataset provides object detection annotations, a few works [3, 103, 98] utilize these annotation for better classification. These object detection annotations are beneficial to noun classification. In our framework, we do not use any additional annotations. We compare to methods that utilize object detection annotations in Table 4.3. Though we do not leverage object detection annotation, we obtain comparable results to SAP [98]. Note that SAP [98] not only incorporates

Methods	Mean Class Accuracy			
	Split1	Split2	Split3	Avg
EgoIDT+Gaze [61]	42.55	37.30	37.60	39.13
I3D (joint) [6]	55.76	53.14	53.55	54.15
I3D+Gaze [59]	53.74	50.30	49.63	51.22
I3D+EgoConv [86]	54.19	51.45	49.41	51.68
Ego-RNN-2S [90]	52.40	50.09	49.11	50.53
LSTA-2S [89]	53.00	-	-	-
Mutual Context-2S [44]	55.70	-	-	-
Prob-ATT [60]	56.50	53.52	53.58	54.53
Prob-ATT+Gaze [60]	57.20	53.75	54.13	55.03
IPL I3D	**60.15**	**59.03**	**57.98**	**59.05**

Table 4.4 : The comparison with the state-of-the-art methods on the EGTEA dataset.

object detection features, but also introduces complicated attention mechanisms with more additional parameters. The results demonstrate the effectiveness of our simple framework.

Results on EGTEA

EGTEA [59] provides gaze and hand mask annotations which have been used by the state-of-the-art methods to provide strong supervision on spatio-temporal attention. EgoIDT+Gaze [61] and I3D+Gaze [59] proposed to utilize gaze point to locate and select the discriminative features. I3D(joint) [6] is a strong baseline on the EGTEA dataset which joint optimizes the two-stream I3D networks. I3D+EgoConv [86] encode head motion and hand masks and further fuse the stream with two-stream I3D model. Prob-ATT [60] is a recent work on EGTEA. With the

Method	Act@1	Verb@1	Noun@1
Indep.	24.1	58.4	33.9
PL	25.4	60.7	35.5

Table 4.5 : Ablation studies on the interactive prototypes on the EPIC-KITCHENS-55 validation set.

gaze supervision during training, Prob-ATT can achieves high recognition results. For fair comparisons, we also utilize the I3D backbone with two-stream architecture to implement out IPL framework. As shown in Table 4.4, our IPL I3D outperforms the I3D(joint) baseline and the state-of-the-art methods by a clear margin on all three splits, even though we do not utilize the gaze annotations and hand mask annotations.

4.3.4 Ablation Studies

The interactive prototypes In our interactive prototype learning framework, the verb prototypes are utilized to assign features for noun classification. To investigate the effectiveness of the interactive learning, we set up a baseline model that uses the independent centers for feature assignment and all the feature groups are used for noun classification. As shown in Table 4.5, our IPL achieves higher results on both verb and noun classification than the model with independent centers. It demonstrates that our interactive learning scheme can not only generate more discriminative feature groups for noun classification but also enhance the verb prototypes to improve verb classification.

The Top-K group selection We design a verb-to-noun group selection approach to disentangle the features from the active object and the features from the distracting object. The feature group selection for noun classification is based on Top-K

Method	Act@1	Verb@1	Noun@1
IPL w/o BG	24.2	60.9	33.5
IPL	25.4	60.7	35.5

Table 4.6 : Ablation studies on the background prototype on the EPIC-KITCHENS-55 validation set.

K	1	3	5	7	15
Noun@1	34.7	34.9	35.5	34.2	33.9

Table 4.7 : Ablation studies on the Top-K selection on the EPIC-KITCHENS-55 validation set.

verb predictions. We conduct experiments to investigate the impact of K on Noun classification. As shown in Table 4.7, the top-1 noun accuracy fluctuates in a small range when K is small. When we continue to increase K to 15, the performance of noun recognition drops by 1.6%. These results show that our IPL is not sensitive to K changes in a proper range. And a too large K would lead to harmful noisy information for noun classification.

The background prototype. We conduct the experiment without the background prototype. As shown in Table 4.6, the top-1 accuracy of noun recognition drops by 2.0%. This result demonstrate that introducing a background prototype can reduce the noise information in the selected noun features.

4.3.5 Qualitative Results

For each input video, K noun feature groups are selected based on Top-K verb predictions for noun classification. Thus, for each feature vector in the spatio-

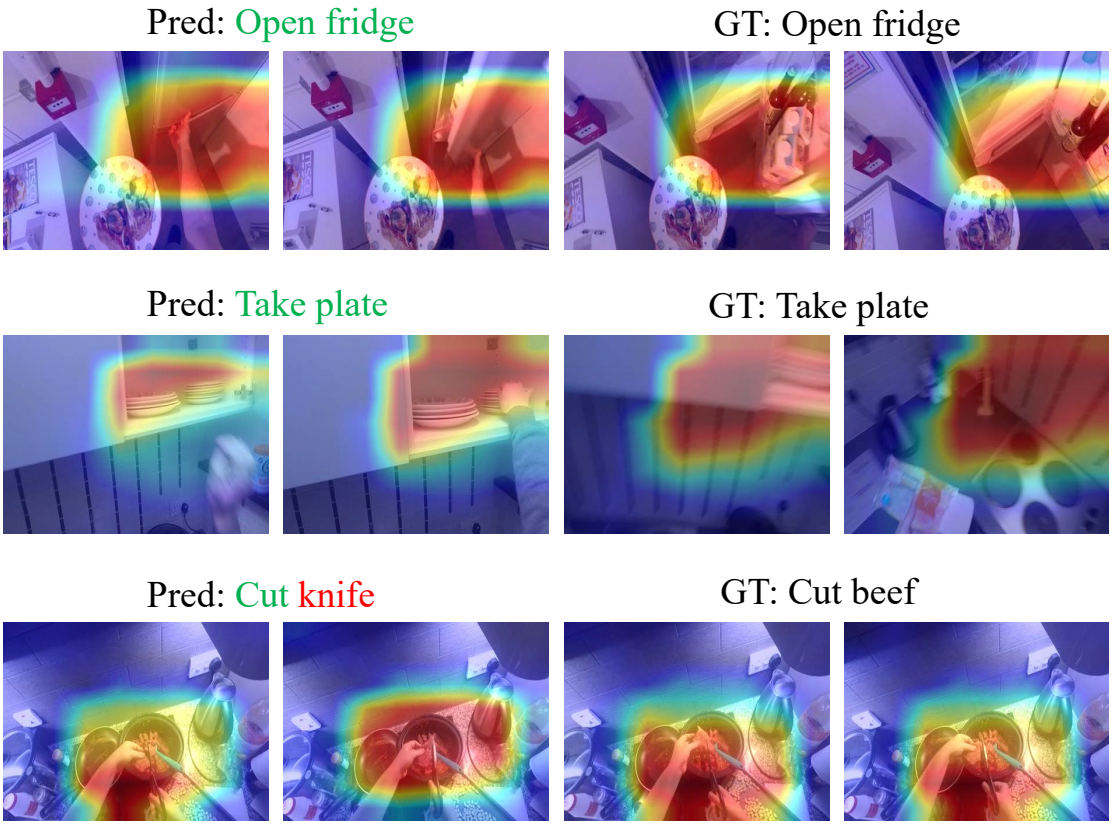

Figure 4.3 : Qualitative results of our IPL model. We illustrate the sum of assignments on the top-K verb prototypes for each feature vector on the spatio-temporal feature map. For each input clip, we uniformly sample four frames and plot the corresponding assignment map. Higher assignment values shows in red. We also print the predicted label and the ground-truth label above the images (Green for correct predictions and Red for failure cases).

temporal feature map, the assignments on the K verb prototypes determine its contributions to the final noun classification. We sample four frames corresponding to the final four spatial feature maps and plot the sum of assignments on Top-K verb prototypes in Fig. 4.3. The text on the top indicates the predicted labels from our model and the ground truth labels. As shown in the fig, the region with high assignment weights indicates the interacting area and the location of the active

objects. These qualitative results explain why our IPL can improve the recognition accuracy of active objects.

In the first two rows, our IPL model produces correct predictions. In the last row, the IPL model fails to predict the active object or the action. This is because the active objects are too small and occluded by the hands. The IPL model can locate the active object and the interacting area, but its robustness based on noisy visual information still needs to be improved.

4.4 Summary

In this Chapter, we present an interactive prototype learning (IPL) framework for egocentric video classification. IPL introduces a set of discriminative verb prototypes to enhance the interactions between noun and verb classification. We evaluate IPL on three large-scale egocentric video classification datasets. Experimental results demonstrate that IPL is able to effectively learn action-centric noun representations. In the future work, we will take a closer look into the affordance of the active object and make full use of this property for better noun recognition.

Chapter 5

Parallel Sampling Network: A Differentiable Framework with Context Relation Mining for Efficient Video Recognition

In the previous Chapters, I introduce two frameworks for egocentric action recognition. And these two video recognition systems both take successive video frames as input, which leads to expensive computational cost. In the real-world applications, we require the video recognition systems to be light and fast. To further improve the efficiency, I investigate into the video frame sampling task in this Chapter. We utilize a light-weight CNN to sample the most salient frames and feed these frames to the heavy-weight video recognition model. The cooperation between these two models not only avoids the expensive computation cost but also improves the recognition accuracy. We demonstrate that the dual-stream modeling on different visual backbones can benefit the overall video understanding system.

5.1 Introduction

In this Chapter, we design a differentiable Parallel Sampling Network for efficient video frame sampling. Video recognition has attracted extensive attention in the past few years due to its wide applications, such as social media analysis, surveillance systems, and robotics. In previous years, huge research attention has been spent on video analysis. By utilizing the latest deep convolutional neural networks (CNNs) with high capacities as the backbone, [6, 125, 85, 97, 92, 57] have boosted the recognition accuracy on videos to high stages. However, in the testing phase, most methods use a classifier on densely sampled frames or clips and aggregate their

predictions to obtain the final recognition scores. This strategy leads to tremendous computational cost and limits the real-world applications of these video recognition systems. Moreover, in real scenarios, most videos are untrimmed and usually longer than that of a trimmed video, which makes the computational cost high if without a carefully designed process mechanism. Besides, there might be a number of frames not relevant to the target label in the untrimmed video. Utilize those irrelevant frames inevitably brings noisy information into the learning process and therefore deteriorates the final performance.

In order to boost the efficiency and accuracy of video analysis, it is necessary to design a mechanism to select the fine discriminative frames from videos. This selection mechanism has advantages in two aspects: 1) it decreases the computational cost since not all, but only a few frames are feed into networks for processing; 2) the noisy or irrelevant frames are discarded in early stages, and therefore their negative effects to the final performance can be negligible. For this purpose, different ways for frame selection in videos are proposed in [53, 104, 108, 21, 115, 97]. [97] manually designs a sampling strategy to select frames from given videos. [53] proposes a video sampler that regards the max classification logits produced by a lightweight CNN as the salience score of the video clip. Then the clips with the top scores are selected to feed a heavy classification model. Although those standalone sampling strategies [97, 53] achieve progress in previous works, they are criticized for their limited generalities. Inspired by the success of Reinforcement Learning (RL), some works [104, 21, 115, 108] treat the video sampling task as a sequential decision problem and introduce RL to optimize the model to select representative frames. However, due to the sequential frame selection strategy of these methods, the global temporal relations among the frames are largely ignored. Moreover, time-consuming recurrent operations prevent them from real-world applications.

To address the problems, we propose a parallel video sampling network (PSN)

with differentiable feedback. Given an input video, we first utilize a pre-sampling strategy to sample a subset of video frames. Subsequently, the proposed PSN can process them efficiently in a parallel manner to generate salient scores. We propose to exploit the inter-relation among the observed frames in the PSN framework, which allows our sampler to select frames based on a comprehensive inspection of the entire video. Experiments demonstrate simple instantiations of the context relation mining module can significantly improve video recognition accuracy. Unlike previous works based on fixed sampling strategies by hand-crafted, our sampler can take advantage of the video-level category signal from the deep recognition model to update itself. Therefore, our framework can encourage the sampler to adapt to different deep recognition model.

We conduct extensive experiments on three challenging video recognition benchmarks: ActivityNet [5], FCVID [47] and Mini-Kinetics [6]. Our framework achieves higher efficiency and better performance than the state-of-the-art video sampling methods. In this work, our main contributions can be summarized as follows:

(1) We develop a simple and effective sampling framework for video recognition. Comparing to previous works, our PSN can sample the relevant frames in a parallel manner and can be updated by receiving differentiable feedback from a subsequent deep recognition model.

(2) We propose to exploit the comprehensive relation among candidate frames, significantly benefiting the sampling procedure.

(3) Our framework consistently outperforms previous works on three large-scale video recognition benchmarks, keeping a better trade-off between accuracy and efficiency.

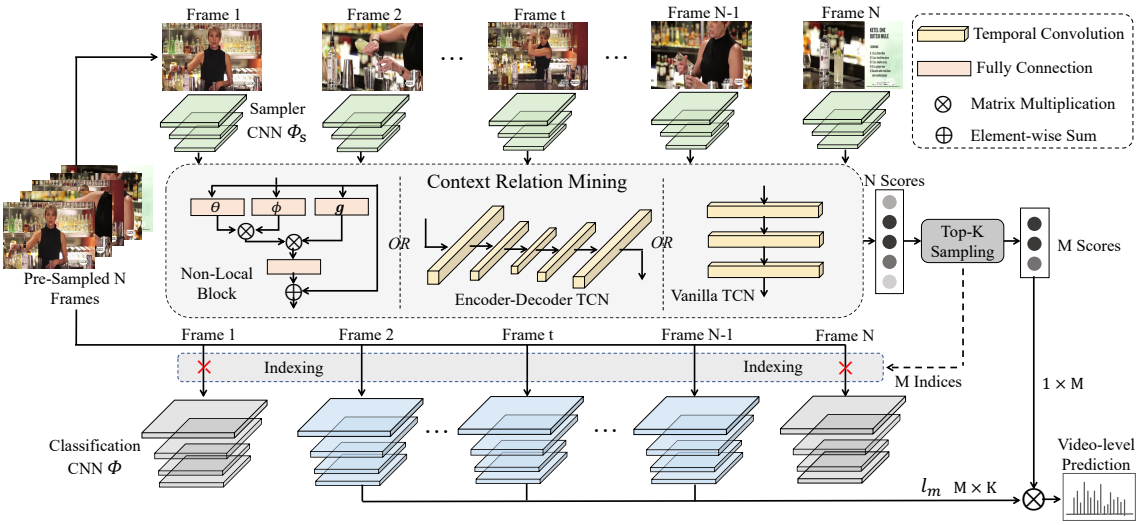

Figure 5.1 : An overview of our proposed parallel sampling network. Given an input video, we pre-sample N candidate frames. The sampler CNN processes the N frames in a parallel manner. The features are fed into a Context Relation Mining (CRM) module to produce N importance scores. We illustrate three instantiations of the CRM module: Non-local Block, Encoder-Decoder TCN, and Vanilla TCN. After that, we utilize the Top-K sampling strategy to select the highest M scores and their corresponding frame indices. The classification model only takes the sampled M frames as input and produces M prediction vectors. Finally, the prediction vectors are multiplied by the selected M weights and averaged as the final prediction.

5.2 The Proposed Approach

5.2.1 Problem Setting

We first illustrate our problem setting. We denote an input video as $V = \{I_1, I_2, ..., I_t, ..., I_T\}$, where I_t denotes the frame at the time slot t and T is the total number of frames in the input video. The video recognition task aims to classify the input video into K classes. Generally, a frame-level or clip-level classifier $\mathbf{\Phi}(\cdot)$ is utilized to produce the prediction probabilities of the frames or clips. To obtain the final prediction score for the entire video, a consensus function $\mathcal{F}$, *e.g.*, average

pooling, is utilized to aggregate the prediction logits of each frame or clip, which is formulated as $\mathcal{F}(\{\mathbf{\Phi}(I_t)\}_{t=1}^{T})$. In this work, we propose a simple and effective method to sample informative frames with length M, *i.e.*, $\{I_{i_1}, I_{i_2}, ..., I_{i_M}\}$, from the original frames, where $M \ll T$. These sampled frames, containing as much discriminative information as possible, can achieve competitive or even better recognition performance with a much lower computational cost.

5.2.2 Parallel Video Sampling Network

Overview

In this section, we illustrate our framework for efficient video recognition. As shown in Fig. 5.1, given an input video, we first pre-sample N candidate frames. This operation aims to maintain the diversity in the candidates set and reduce the difficulty of modeling long sequences. Besides, the global pre-sampling strategy allows us to model the comprehensive structure of the input video. After that, a lightweight CNN extracts features of the N frames swiftly in a parallel manner. A context relation mining (CRM) module is introduced to explore the inter-relation among these frames, which is crucial to the sampling procedure. The outputs of the CRM module are fed into a fully-connected layer, followed by a sigmoid activation function to generate N sampling scores. These scores measure the importance of the corresponding frames for the video-level prediction.

In the training phase, the classification model takes all N sampled frames as input and produces N prediction vectors. These vectors are multiplied by the N scores and then aggregated as the final prediction. The prediction is utilized to calculate the classification loss. A sparsity regularization loss is applied to the generated scores. It is used to enforce the number of frames with high values converge to the expected M. The gradients of the two losses are back-propagated to optimize the sampler while the weights of the classification model are frozen. Therefore, given

an arbitrary recognition model, the sampler can be optimized adaptively with our framework.

In the testing phase, the sampler net processes N candidate frames and generates N scores. We select the highest M scores and the corresponding frames using a Top-K sampling strategy. The classification network only takes M sampled frames as inputs to reduce computational cost. The final video-level prediction is obtained by a weighted sum of the selected M weights and M frame-level predictions.

Pre-sampling

Most existing video sampling methods [21, 104, 115] assess the observed frames sequentially. This strategy limits the observation range of context frames. In these scenarios, the sampling of the next frame is only based on the historical frames. Besides, the process of the next frame should wait for the processing of the current frame, which diminishes the efficiency of the sampler compared to parallel inference.

To avoid these problems, we propose to pre-sample N candidate frames from the input video. Given a video V, we divide it into N segments $\{S_1, S_2, ..., S_N\}$ of equal duration. In the training phase, one frame is randomly sampled from its corresponding segment S_k. In the testing phase, the center frame of each segment is selected. This sampling strategy ensures the sampled frames to distribute uniformly along the temporal dimension, which covers the visual content of the whole video. Therefore, we can explore the global structure of the input video and select the most important frames based on the comprehensive observation. Moreover, all the N frames can be sampled at one time and processed in a parallel manner, which ensures the efficiency of our framework. This sampling strategy is first introduced in [97], which feeds the sampled frames to the recognition model directly. In contrast to their method, we feed the sampled frames to the sampler network for efficient video classification. The input to the recognition model is based on the processing

of the pre-sampled frames. Therefore, we require a larger number of segments than the standard recognition task to provide adequate sampling candidates.

Context Relation Mining

After pre-sampling, we obtain N candidate frames, which covers the visual content of the whole video. We aim to sample the most informative M frames from the subset. The importance sorting is determined by the comparisons among candidate frames, where frame correlations need to be well explored.

We should investigate the frame correlations based on a comprehensive inspection of the entire video rather than only considering the salience of independent frames. It is crucial to exploit the context relation to sample important frames. Therefore, we propose to integrate a context relation mining module to our efficient sampling framework. Considering both efficacy and efficiency, we expect the context relation mining module to meet the following requirements: (1) it is capable of modeling both the local and global relation among frames; (2) it needs to process all the frames in a parallel manner. The implementation of the CRM module is flexible. We will discuss multiple instantiations in the next subsection.

Formally, given a pre-sampled subset $\{I_1, I_2, \ldots, I_N\}$, the lightweight CNN $\boldsymbol{\Phi_s}$ encodes the frames into N features. After that, our CRM module can exploit the context relation to generate an enhanced feature matrix $\boldsymbol{p}$:

$$\boldsymbol{p} = \mathcal{CRM}(\boldsymbol{\Phi_s}(I_1), \boldsymbol{\Phi_s}(I_2), \ldots, \boldsymbol{\Phi_s}(I_N)), \tag{5.1}$$

where $\boldsymbol{p} \in \mathbb{R}^{N \times C}$, C is the feature dimension, and $\mathcal{CRM}$ denotes the CRM module. The enhanced features are used to generate an importance score vector by a fully-connected layer with parameter $\boldsymbol{W_s}$ and a `Sigmoid` activation function:

$$\boldsymbol{q} = \texttt{Sigmoid}(\boldsymbol{p}\boldsymbol{W_s}), \tag{5.2}$$

where $\boldsymbol{W_s} \in \mathbb{R}^{C \times 1}$, the output score vector $\boldsymbol{q} \in \mathbb{R}^{N \times 1}$.

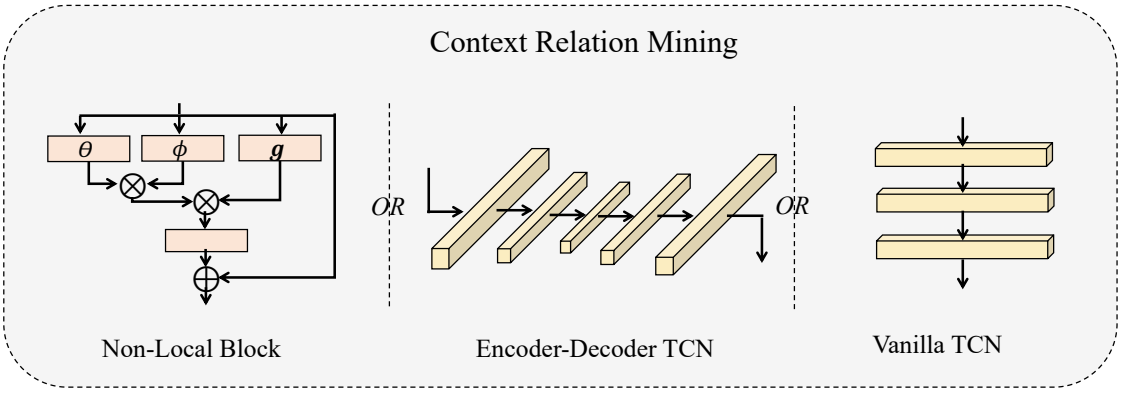

Figure 5.2 : Context Relation Mining Module Instantiations.

Context Relation Mining Module Instantiations

The CRM module can be implemented in a variety of ways. We experiment with the following choices: **(1) Non-local Block.** Non-local Neural Network [100] was proposed to model the spatio-temporal relations in videos. It is a variant of the self-attention mechanism [95]. We can utilize non-local blocks to build our CRM module. Specifically, given a set of candidate features extracted by the lightweight CNN, each input frame feature is used to attend all the candidate features and generate dot-product similarities. All the features are then aggregated with the corresponding similarity weights for each query feature to provide temporal dependencies. **(2) Vanilla Temporal Convolution Network.** Temporal convolution network utilizes 1-D convolution on the temporal dimension to model the temporal dependency. In the vanilla version of TCN, we use three stacked temporal convolution layers with fixed kernel size to build the CRM module. The hidden states produced by the first layer has the same size as the input sequence length. Compared to the Non-local block, the learned temporal convolution kernel can mine the pair-wise transitions between consecutive frames, which is crucial for the frame sampling task. However, due to the limitation of the receptive field of the vanilla TCN, the long-term dependency may be neglected. Thus, we design a variant of

TCN in the following to alleviate this problem. **(3) Encoder-Decoder Temporal Convolutional Network.** We adopt a modified Encoder-Decoder TCN [58] as the context relation mining module in our sampler. Both encoder and decoder parts consist of 3 layers and we add shortcut connections between the decoder and the encoder. The temporal resolution is first reduced to $\frac{1}{8}$ of the input size and then up-sampled to the same size as the input features. This design encodes the rich global context to the intermediate state which benefits the subsequent score generation. The skip-connection encourages the module to make use of both low-level and high-level features. In the encoder part, each layer is implemented with a 1D convolution, batch normalization, and ReLU activation function. In the decoder part, we reverse the structure of encoder and replace the 1-D convolution with 1-D deconvolution, and each deconvolution layer has the same number of filters as the corresponding convolution layer in the encoder. The input for each decoder layer is the element-wise sum of the output of the last layer and the feature map from the encoder. Compared to vanilla TCN, the Encoder-Decoder TCN has larger receptive field and can take advantage of the global context information.

5.2.3 Training Objectives

The classification loss

In the training phase, N frames $\{I_1, I_2, \ldots, I_N\}$ are pre-sampled to be fed to the sampler. The classification model, denoted as $\mathbf{\Phi}$, encodes the frames to N logits:

$$\boldsymbol{l_m} = (\mathbf{\Phi}(I_1), \mathbf{\Phi}(I_2), \ldots, \mathbf{\Phi}(I_N)), \tag{5.3}$$

where $\boldsymbol{l_m} \in \mathbb{R}^{N \times K}$ and K denotes the number of target classes. The output logits of the classification model are multiplied by the scores generated by the sampler ($\boldsymbol{q} \in \mathbb{R}^{N \times 1}$). Then we average the weighted logits of N frames and apply a `Softmax` function to normalize the logit distribution to obtain the final video-level prediction

$\boldsymbol{l_c}$. Formally, $\boldsymbol{l_c} \in \mathbb{R}^K$ is calculated by:

$$\boldsymbol{l_c} = \texttt{Softmax}(\boldsymbol{q}^{\mathrm{T}}\boldsymbol{l_m}). \tag{5.4}$$

Let $\boldsymbol{y} \in \mathbb{R}^K$ denotes the one-hot or multi-hot ground-truth video label, the classification loss is then represented as the cross-entropy between $\boldsymbol{l_c}$ and $\boldsymbol{y}$:

$$\mathcal{L}_c = -\mathbb{E}[\boldsymbol{y}^T log(\boldsymbol{l_c})]. \tag{5.5}$$

The weights of the classification network are pretrained. They are fixed during the sampler training. We only need to optimize the sampler to minimize $\mathcal{L}_c$, which enforces the sampler to generate appropriate scores based on the logits $\boldsymbol{l_m}$. This objective encourages the sampler to learn the sampling strategy at video-level with the feedback of the classification model, which is beneficial to the prediction of the entire video.

The sparsity regularization loss

Moreover, we add a regularization loss for the sampler to increase the diversity of the output weights. The regularization loss for the sampler is formulated as:

$$\mathcal{L}_r = max(\sum_{i=0}^{N} q_i - M,\ 0), \tag{5.6}$$

where q_i denotes the element in the score vector $\boldsymbol{q}$, M is a margin which is set to the number of sampled frames. Minimizing this loss function can enforce the scores for uninformative frames close to zeros. Without the regularization loss, the sampler can easily collapse and output ones for all frames. The margin M can avoid all the output scores being zeros. When the sum of the scores is less than M, this regularization term will have no effect on the optimization.

The overall loss

We combine the two losses to yield our final objective for the sampler:

$$\mathcal{L} = \mathcal{L}_c + \lambda\mathcal{L}_r, \tag{5.7}$$

where λ is a hyper-parameter to control the weight of the regularization loss. These gradients of the two losses are back-propagated through the generated score vector $\boldsymbol{q}$ to optimize the sampler network.

5.2.4 Inference Strategy

In the training scenario, the classification model processes N frames. While in the testing phase, we only process the most informative M frames for high efficiency because the computational cost of the heavy-weight classification model is the main cost source in our framework. To this end, we develop a Top-K Sampling strategy. We first pre-sample N frames uniformly from the input video. The lightweight CNN processes these frames swiftly in a parallel manner to produce N frame features. Our CRM module exploits the inner-relation among these features and generates scores for each input frame. After that, we select the top-M scores and their corresponding M indices. The classification network can index M frames from the pre-sampled frames and take the selected frames as inputs to produce M frame-level predictions. The final video-level prediction is calculated by computing the weighted-sum of the M predictions with the M scores. The weighting operation can emphasize the most informative frames and enhance the robustness when most frames in the input video are noisy.

5.3 Experiments

5.3.1 Experimental Setup

Datasets

We evaluate our method on three large-scale video recognition datasets.

ActivityNet [5] is a large-scale video benchmark for human activity understanding. We evaluate our method on the second version dataset, i.e., ActivityNet v1.3.

It contains 19,994 videos from 200 activity categories. Notably, the labels of the test set are not publicly available. Thus the performances on the ActivityNet dataset are all reported on the validation set. We decode videos at 1 FPS and use RGB frames only in our experiments considering efficient computing.

FCVID [47] is a challenging video dataset for untrimmed video classification, which consists of 91,223 web videos annotated manually according to 239 categories. The average duration of videos in FCVID is 167 seconds and the dataset is split into a training set with 45,611 videos and a testing set with 45,612 videos. Compared to the action-oriented videos in ActivityNet, the video classes in FCVID are more generic such as social events (*e.g.*, "tailgate party"), objects (*e.g.*, "panda") and scenes (*e.g.*, "beach"). Videos in FCVID were decoded at 1 FPS.

Mini-Kinetics [71] is a subset of the large-scale trimmed video dataset Kinetics [6]. We use the split in ARNet [71] to sample 200 classes and 131,082 videos from Kinetics. The average duration of these videos is 10 seconds and videos are decoded at 30 FPS.

Evaluation Metrics

Following the standard setting, we use mean average precision (mAP) as the evaluation metric for ActivityNet and FCVID. For Mini-Kinetics, we report Top-1 accuracy. We utilize GFLOPs to estimate model computational cost. An efficient video recognition system should have low GFLOPs while maintaining high accuracy.

Implementation Details

We adopt ShuffleNet-V2 [70] as the backbone in the sampler. The sampler takes 25 frames of size 224×224 as input. For the recognition model, we train the ResNet [38] models with two different strategies for a fair comparison. One scheme

utilizes the TSN strategy [97] to train the recognition model. The other scheme takes independent video frames to train the model, following [108]. The TSN model can achieve higher recognition accuracy than the frame-based model. Random scaling and cropping are used at the training stage, and only center-cropping is applied at the inference stage. The hyper-parameter λ for the sparsity regularization loss is set to 0.1.

Sampler details We adopt ShuffleNet-V2 [70] as our lightweight CNN in the sampler. The CRM is implemented by temporal convolution and deconvolution with a stride of 2 and the kernel size is 3. In this training phase, we train the sampler net for 15 epochs. We set the initial learning rate at 0.002 and the learning rate is set to 0.0002 at the 10th epoch. We utilize SGD optimizer with a momentum of 0.9 and a weight decay of 1.0×10^{-4}.

Recognition model details For the recognition model, we pre-train the model with two different strategies. The first one is utilizing the method in TSN [97] to train the model. The second one is adopted in [108], which takes independent frames in the videos to train the network. We adopt ResNet-50, ResNet-101 and ResNet-152 [38] pretrained on ImageNet [15] as our backbones. We utilize SGD optimizer with a momentum of 0.9 and a weight decay of 1.0×10^{-4}. We train the model for 50 epochs and the learning rate is decayed by a factor of 10 every 20 epochs. For TSN training strategy, the batch size is 32 and the initial learning rate is set to 0.002. For frame-based training strategy, we train the model with a batch size of 64 and the initial learning rate is set to 0.001.

Method	mAP
TSN Baseline	79.06
Sampler w/o CRM	78.93
Sampler w/ Non-local Block	78.44
Sampler w/ Vanilla TCN	82.01
Sampler w/ ED-TCN	**82.28**

Table 5.1 : Ablation study on CRM on ActivityNet.

5.3.2 Ablation Studies

Effectiveness of Context Relation Mining

We propose Context Relation Mining (CRM) to introduce context information and exploit temporal relations among frames in the video. We discuss several instantiations of the CRM module, such as Non-local Block, Vanilla TCN, and Encoder-Decoder TCN. We use TSN-based ResNet-101 as the recognition model and evaluate these sampler variants on the ActivityNet dataset. The results are illustrated in Table 5.1.

"TSN baseline" denotes uniform sampling of ten frames for the recognition model in the testing. To demonstrate the effectiveness of context modeling, we train the sampler without CRM ("Sampler w/o CRM"). This model achieves 78.93% on mAP. Compared to the best model with CRM, the performance drops by 2.97%. "Sampler w/o CRM" fails to outperform "TSN Baseline", which demonstrates that the context relation is crucial for frame sampling. The best performance is achieved by the model "Sampler w/ ED-TCN". It is implemented by an Encoder-decoder TCN with shortcut connections. This model outperforms the "TSN Baseline" by 3.22% on mAP. Moreover, we evaluate another two variants of CRM, *i.e.*, "Sampler w/ Non-local Block" and "Sampler w/ Vanilla TCN". "Sampler w/ Vanilla TCN" indicates

λ	0	0.01	0.04	0.06	0.10	0.15
mAP	78.72	80.37	81.90	82.18	**82.28**	82.25

Table 5.2 : Results of different λ on ActivityNet.

Method	Sampling	Weighting	mAP
TSN 10 frames	-	-	79.06
Sampling 10 frames	✓	-	81.70
Weighting 10 frames	-	✓	80.00
Weighting sampled 10 frames	✓	✓	**82.28**

Table 5.3 : Ablation study on the weighting and sampling operation on ActivityNet.

that the sampler leverages stacked temporal convolutions as CRM. Our best model outperforms this variant. This is because the encoder-decoder structure can capture more extended relations among all observed frames, which benefits the sampling procedure. Besides, we utilize a Non-local block [100] to model the relation instead of temporal convolutions, as indicated by "Sampler w/ Non-local Block". The variant fails to boost the performance over the baseline. It is because the non-local operation overlooks the relative location information among orderly frames.

In the following, we conduct experiments with the best model. "Our PSN" in the following tables leverages ED-TCN in context relation modeling.

Analysis on the weight of the sparsity regularization loss

We add a regularization loss $\mathcal{L}_r$ to enforce the sampler to generate sparse scores, which is important for the subsequent frame selection and final classification. In Table 5.2, we report the impact of the weight for the regularization term. The experiment is based on the TSN ResNet-101 recognition model. When $\lambda = 0$, the

Method	FCVID		ActivityNet	
	mAP	GFLOPs	mAP	GFLOPs
Uniform Sampling	80.4%	195.8	70.2%	195.8
FastForward	67.6%	66.2	54.7%	17.2
FrameGlimpse	71.2%	29.9	60.2%	32.9
AdaFrame	80.2%	75.0	71.5%	79.5
MARL	-	-	72.8%	375.8
Listen2Look (Image)	-	-	72.3%	81.3
LiteEval	80.0%	94.3	72.7%	95.1
SCSampler*	80.5%	82.3	72.9%	82.3
Our PSN	**81.1%**	82.3	**73.5%**	82.3

Table 5.4 : Comparison with the state-of-the-art sampling strategies on FCVID and ActivityNet. We train the recognition model using **frame-based** training strategy following [108]. * indicates our implementation.

sampler is optimized only with the cross-entropy loss $\mathcal{L}_c$. The result is lower than the TSN baseline. It is evident that the regularization term for the sampler is indispensable. The mAP is first improved with the increase of the weight and the performance is stable when $\lambda > 0.1$. This is because we set a margin in the regularization loss. When the sum of the scores is lower than the number of sampled frames, the loss is equal to zero. This design can alleviate the damage of the regularization loss when the model has found the optimal sampling strategy. According to the above analysis, we set $\lambda = 0.1$ in our experiments.

Analysis on the sampling and weighting operation

Our final video-level prediction is produced by computing the weighted sum of the M predictions with the M scores. We evaluate the influence of the weighting opera-

Method	Arch	FCVID		ActivityNet	
		mAP	GFLOPs	mAP	GFLOPs
Uniform Sampling	R-101	82.2%	195.8	81.3%	195.8
SCSampler*		81.3%	82.3	79.9%	82.3
MARL		-	-	81.5%	837.8
Our PSN		**82.8%**	**82.3**	**82.3%**	**82.3**
ARNet	R-50	81.1%	35.1	73.8%	33.5
Our PSN		**81.5%**	36.8	**78.3%**	**28.6**

Table 5.5 : Comparison with the state-of-the-art methods based on **TSN** recognition model [97] on FCVID and ActivityNet. * indicates our implementation.

tion and the sampling operation on ActivityNet. The results are shown in Table 5.3. "TSN 10-frames" indicates the recognition model processes the 10 frames using the TSN sampling strategy instead of our sampler network. "Sampling 10 frames" indicates selecting the top-10 frames by the sampler net and averaging their prediction vectors as the final video prediction. "Weighting 10 frames" indicates directly training the sampler to generate 10 weights for 10 input frames without sampling, and the video-level prediction is computed by the weighted sum of 10 frame-level predictions. "Weighting sampled 10 frames" computes the weighted-sum of the predictions from sampled top-10 frames. We can observe that the sampling operation plays the most important role in our framework. "Sampling 10 frames" outperforms "TSN 10 frames" by a large margin without weighting operation. Comparing "Weighting 10 frames" with "TSN 10 frames", we observe that the weighting operation can boost the recognition mAP by 0.94%.

5.3.3 Comparison with the State-of-the-Art

Comparison with other frame sampling methods.

We evaluate our method on both FCVID and ActivityNet dataset. We compare with the following standalone sampling methods: **Uniform Sampling** is a hand-crafted sampling strategy; **SCSampler** [53] samples frames based on the largest confidence scores from the lightweight classifier. **MARL** [104], **FastForward** [21], **FrameGlimpse** [115] and **AdaFrame** [108] are designed based on Reinforcement Learning algorithms to select pre-defined actions to skip frames. **LiteEval** [107] is optimized to skip frames with gumbel-softmax trick. **Listen2Look** [27] utilizes sequential attention mechanism to select the next processing frame. **ARNet** [71] learns to adaptively select different input resolutions for different frames.

Our Parallel Video Sampling Network (PSN) outperforms all sampling methods on two large-scale datasets. For fair comparisons, we train the recognition models with the same settings as the state-of-the-art methods, and we implement SCSampler using the same lightweight CNN as the sampler like our PSN. In Table 5.4, we illustrate the results with frame-based ResNet-101. In Table 5.5, we show the results using TSN-based ResNet-50 and ResNet-101. The results of FrameGlimpse and FastForward are quoted from the reproduction in [107]. Compared to the Uniform Sampling baseline, our method achieves higher mAP while reducing the computing complexity more than half on both two datasets. This demonstrates the efficacy and efficiency of the proposed sampling framework. FrameGlimpse and FastForward achieve the lowest GFLOPs but the recognition performance is much lower than our method. This is because these two sequential sampling approaches skip lots of important frames in the videos. In contrast, our PSN achieves a balance between speed and accuracy. Our method outperforms SCSampler on ActivityNet by 2.4% with TSN-based classifier and 0.9% with frame-based classifier. SCSampler fails to

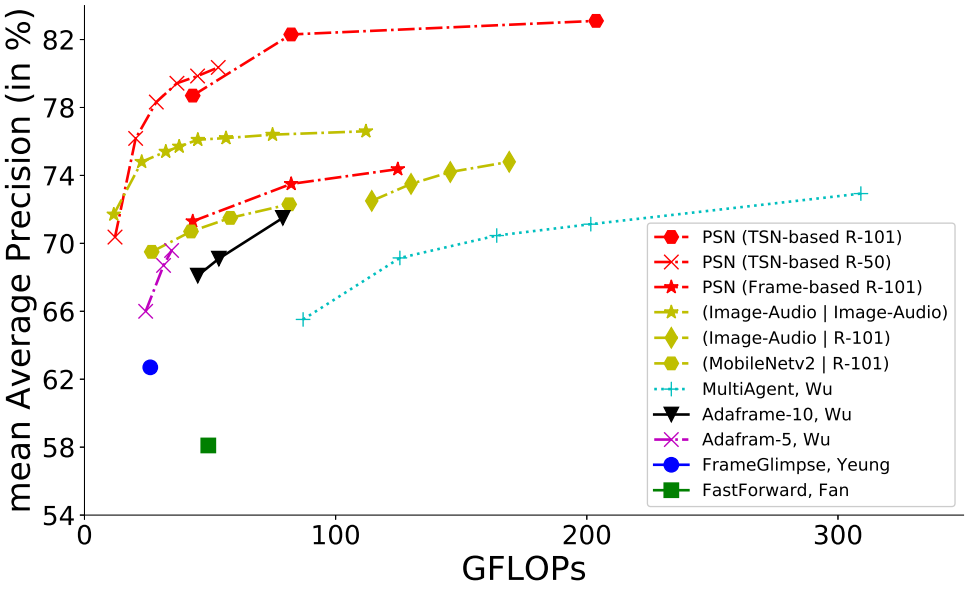

Figure 5.3 : Mean Average Precision vs. Computational Cost on AcitivityNet. Comparison with state-of-the-art methods.

outperform the uniform sampling baseline with TSN-based classifier. This demonstrates our method is more adaptive than the standalone sampling strategy. Our PSN achieves higher recognition accuracy than other sequential sampling methods. This demonstrates the effectiveness of our parallel sampling design and the context relation mining module.

In Fig. 5.3, we illustrate the mAP v.s. GFLOPs curves on ActivityNet. We quote the baseline results from [27]. Our PSN achieves clear improvement with the same recognition model, *i.e.*, Frame-based R-101. Moreover, our TSN-based R-50 outperforms existing methods by a large margin. This indicates our method can keep a better trade-off between computational cost and recognition accuracy.

We further evaluate our PSN on Mini-Kinetics. The results are shown in Table 5.6. Our PSN outperforms the recent methods on both Top-1 accuracy and speed, which demonstrates our method is superior on trimmed video recognition.

Method	Top-1	GFLOPs
LiteEval [107]	61.0	99.0
SCSampler	70.8	42.0
ARNet [71]	71.4	32.0
Our PSN	**72.7**	**28.6**

Table 5.6 : Quantitative comparisons on Mini-Kinetics.

Comparison with other video recognition methods

We compare our sampling strategy with different video recognition methods on the ActivityNet dataset. The results and comparisons are summarized in Table 5.7. **P3D** [80] is a deep video recognition model with decomposed 3D convolution. **RRA** [124] proposes to modulate the feature maps and abandons weak activation for untrimmed video recognition. **DSN** [120] is an RL-based video sampling method. We use its RGB stream for a fair comparison. We utilize ResNet-152 as our recognition model, which is trained by the same TSN-based strategy as [104]. Our method outperforms all the previous approaches, which demonstrates the superiority of our sampling-based recognition scheme for video recognition.

5.3.4 Analysis on the sampling scores.

In the testing phase, the frames are selected based on a Top-K strategy. We visualize the relative location of the sampled top-10 frames. As shown in the left of Fig. 5.4, the sampled frames at the beginning and the ending of the videos are much less than the frames at the middle location. This is because the videos usually contain uninformative frames or motion blur when they are close to the start and the end. In our framework, the sampler takes the pre-sampled N frames as input and produces N scores, and then the top-K sampling operation is applied to the N scores

Method	Arch	mAP
P3D [80]	ResNet-152	78.86
RRA [124]	ResNet-152	83.42
DSN-RGB [120]	R(2+1)D-34	83.50
MARL [104]	ResNet-152	83.81
Listen2Look [27]	ResNet-152	84.20
Our PSN	ResNet-152	**84.76**

Table 5.7 : Quantitative comparisons on ActivityNet.

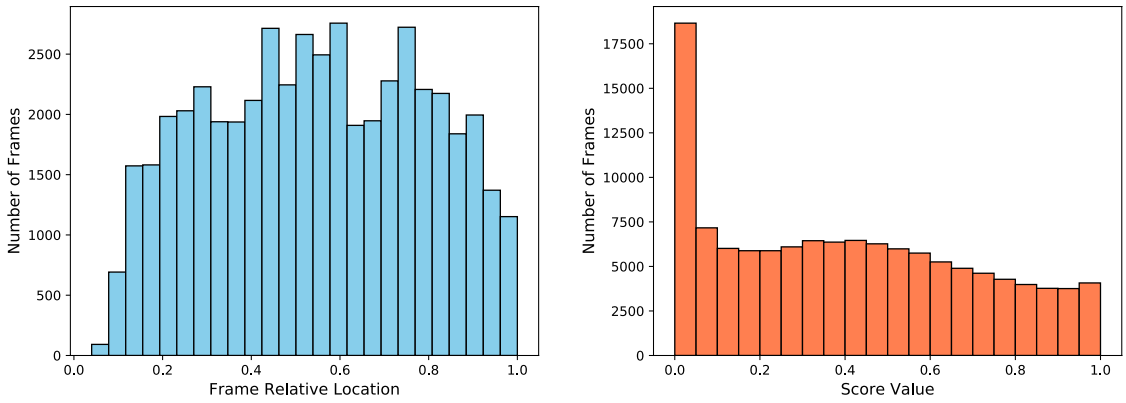

Figure 5.4 : The left is the histogram of relative sampling location. The right is the histogram of scores produced by the sampler.

and frames. We set the number of pre-sampled frames N to 25 and select 10 frames from them. The distribution of the generated scores are illustrated in the right of Fig. 5.4. The scores of a large number of frames are close to zero. This indicates that the sparsity regularization loss and the cross-entropy loss can enforce the sampler to generate very small weights for the frames that need to be abandoned.

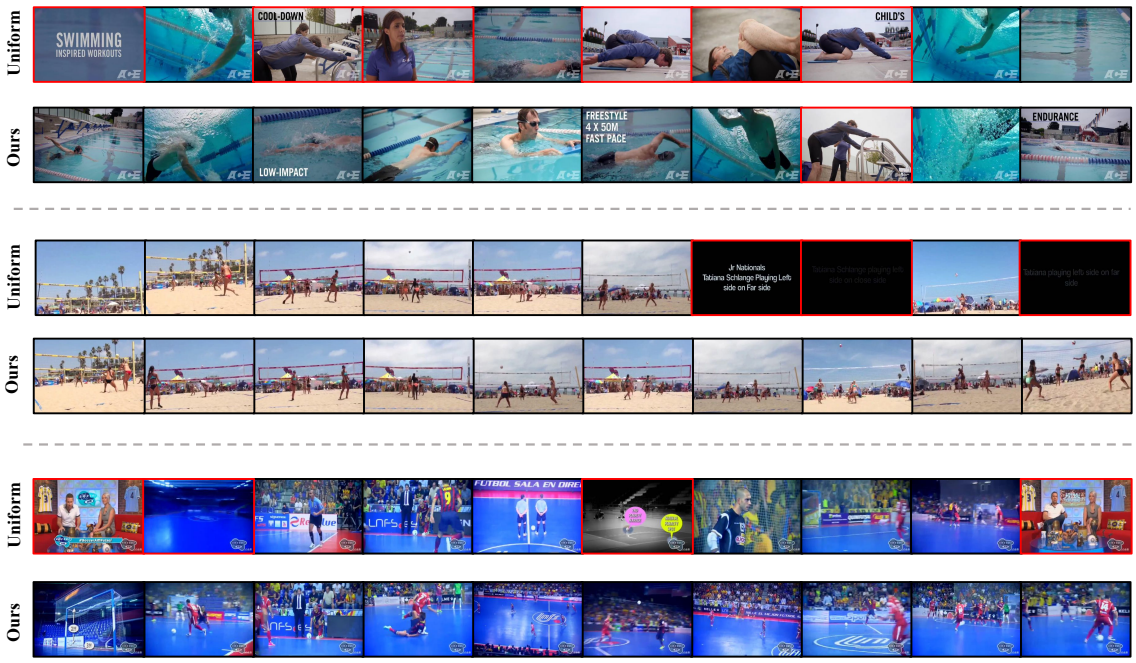

Figure 5.5 : Visualization of the Uniformly Sampled 10 frames and the 10 frames sampled by our PSN. The ground truth actions for the videos are *"Swimming"*, *"Playing beach volleyball"* and *"Futsal"*, respectively. The red box indicates that the frame is irrelevant to the action class empirically.

5.3.5 Qualitative Results

We visualize the frames sampled by our PSN and Uniform Sampling strategy in Fig. 5.5. Our PSN achieves better performance than uniform sampling. For example, in the first row, uniform sampling strategy selects six frames (indicated by red boxes) that are not indicative to the action *"Swimming"*. These noisy frames diminish recognition accuracy. In contrast, the frames sampled by our sampler are more relevant to the corresponding action. Moreover, the frames sampled by our PSN are more salient and contain less motion blur, which benefits to the final video recognition. These visualization results demonstrate the effectiveness of our method.

However, when the length of the input videos is too short, the PSN model samples

some irrelevant frames, which may decrease the recognition accuracy. It is because, in our framework, the number of sampled frames is pre-defined. Thus, the number of candidate frames may be fewer than the number of sampled frames in some short videos. In the future, we will modify our framework to make the number of sampled frames adaptive to the length of the input video.

5.4 Summary

In this Chapter, we presented a simple yet effective video recognition framework. We proposed to train a parallel frame sampler by the feedback from the deep video recognition model. Moreover, we proposed a context relation mining module to model the inter-relation among the candidate frames, which allows the sampler to sample frames based on a comprehensive inspection of the entire video. Several simple instantiations of the CRM module can significantly improve the recognition accuracy. The proposed sampler can process the video in a parallel manner and can be optimized in high efficiency. Our framework consistently outperformed previous works on three video recognition datasets.

Chapter 6

T2VLAD: Global-Local Sequence Alignment for Text-Video Retrieval

Cross-modal analysis is an important part of the video understanding field. In this chapter, I introduce a cross-modal video retrieval framework and demonstrate the spirit of dual-stream modeling is essential for this task. The common practice for the video retrieval task is to map the text description and video content into a shared feature space. To bridge the gap between the language information and video content information, we propose a global-local sequence alignment method.

6.1 Introduction

Recently, the modeling of mulit-modal data has attracted increasing interests as it is beneficial in many real-world applications, *e.g.*, keyword-based image or video retrieval systems. Some efforts have been made in building retrieval system with complex text inputs [8], *e.g.*, retrieving contents of "a group of men inspect and test a brand new yellow car". This is more applicable as the users could search contents based on more detailed descriptions, which allows users quickly locating interested video clips.

One of the promising directions to enable cross-modal video retrieval is to measure text-video similarities using metric learning [111]. In this case, the common practice is to embed both descriptions and videos into a joint embedding space. Different to text-image retrieval, it is more challenging to align texts and video cues due to larger temporal variations in video data. Chen et al. [8] proposed a hierarchi-

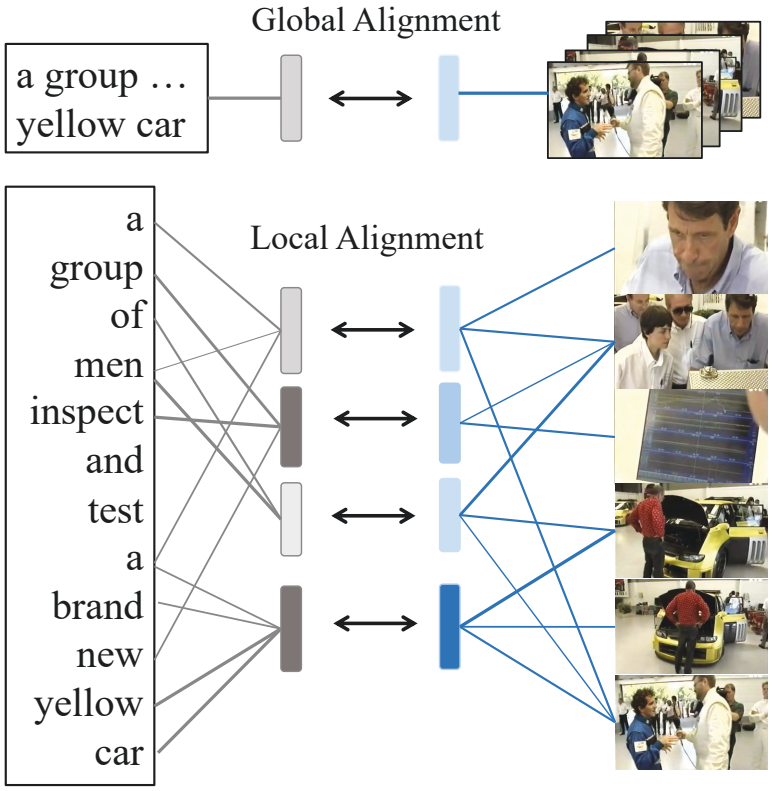

Figure 6.1 : Global and local alignment between texts and videos.

cal graph reasoning model to capture both global events and local actions through a local graph matching. They manually designed three levels of semantics, including events, actions, and entities, while graph convolutional networks are utilized for encoding complex hierarchical text semantics. Others focus on the learning of effective language and video representations. For instance, Gabeur et al. [26] leveraged a multi-modal transformer to gather valuable cross-modal and temporal cues on complex events.

We aim to align text-video sequences in a global-local perspective. Besides the global alignment between text and video data [26], in the **local perspective**, we aim to utilize a number of learnable semantic topics to jointly summarize both text and video data. Instead of parsing text descriptions to a hierarchical semantic role graph [8], it is hoped that these semantic topics could be discovered and automati-

cally learned during the end-to-end training of the cross-modal retrieval model. We could further share the weights of text topics and video topics to offer a joint topic representation learning and to reduce the semantic gap between text and video data. To achieve local alignment, we could minimize the distance between the grouped text feature and the grouped video features.

We implement the idea of local semantic topic alignment with the help of a NetVLAD operation [2]. Vector of Locally Aggregated Descriptors (VLAD) [46] and its learnable variant, *i.e.*, NetVLAD, have been widely used in content-based image retrieval and video retrieval due to their excellent performance in aggregating local features. In NetVLAD, the learnable centers are regarded as "visual words" of the input data, which can readily be utilized as latent semantic topics on our cross-modal video retrieval task. For both text and video modalities, we use NetVLAD operations to obtain an aggregated feature for each topic, where the topic centers are **shared** between the two modalities. The text features and video features are softly assigned to topics based on their corresponded similarities. Instead of applying classification on the global discriminative feature vector [2], we align the locally aggregated text features and video features using a ranking loss.

Without complex graph operations [8] and multi-layer transformers [26], we surprisingly find that our collaborative encoding method, namely T2VLAD, could boost the retrieval performance on various datasets. The contributions can be summarized as below:

- First, we introduce to automatically learn text-and-video semantic topics and re-emphasize the importance of local semantic alignment between texts and videos for better cross-modal retrieval. When matching complex text queries with temporal dynamic videos, the local alignment plays an essential role in jointly locating the text-video cues. This is also the main difference that text-

video retrieval differs from text-image retrieval.

- Second, we introduce an effective strategy to locally align text inputs and video inputs. Based on the success of NetVLAD encoding [2], we propose a T2VLAD encoding for cross-modal retrieval, where we exploit shared VLAD centers to reduce the semantic gap between texts and videos.

- Third, we demonstrate significant improvements of T2VLAD on three standard text-video retrieval benchmarks, *i.e.*, MSRVTT [112], ActivityNet Captions [54], and LSMDC [82]. Notably, we outperform a HowTo100M-pretrained [75] multi-modal transformer [26] with 2.9% gain (Rank@1) on MSRVTT without any additional data.

6.2 Method

6.2.1 Overview

We propose a Text-to-Video VLAD (T2VLAD) for the cross-modal retrieval, which aligns text and video features in a global and local perspective. Given a text-video pair, our goal is to encode it into a joint feature space to measure the similarity. As shown in Fig. 6.2, we leverage multiple experts to extract the local video features corresponding to each modality (Section 6.2.2). The BERT model is utilized to extract contextual word features (Section 6.2.3). After that, we feed all the video features from different experts to a self-attention layer to enhance the features based on cross-modal relations. The output video features and text features are assigned to a set of cluster centers, which are shared between text encoding and video encoding. We aggregate the local features based on the assignments and generate the local aligned features for both video and text to compute a local video-text similarity (Section 6.2.4). To provide additional supervision on the local alignment and introduce complementary information, we develop a global alignment

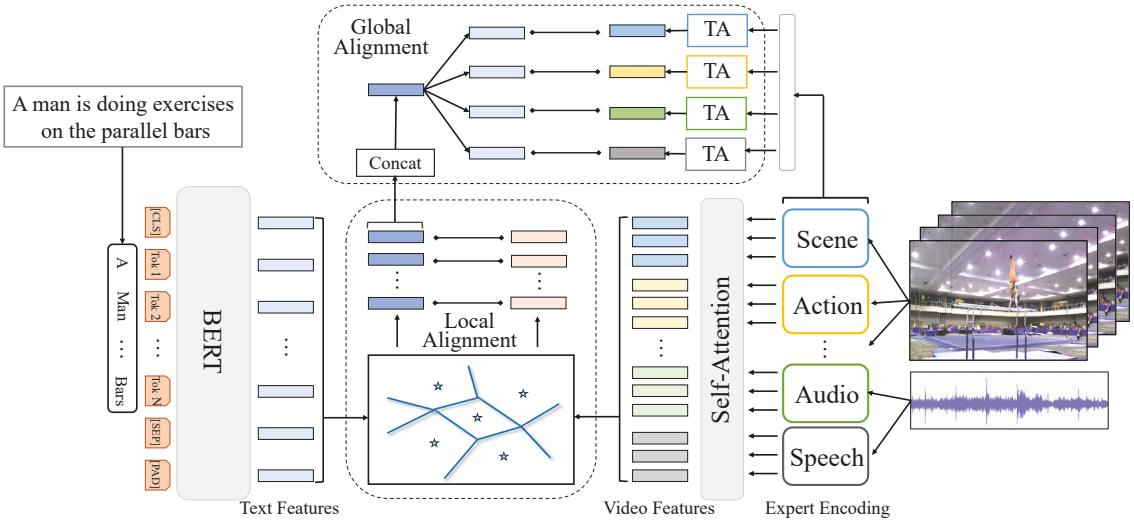

Figure 6.2 : Our T2VLAD framework for text-video retrieval. "TA" is a temporal aggregation method. For simplicity, we use a max-pooling operation to aggregated each expert.

scheme (Section 6.2.5). The video features from each expert are aggregated to a global feature to calculate a similarity with the projected global text feature.

6.2.2 Video Representations

Compared to image data, videos are more complex and contain richer information such as motion, audio and speech. To make full use of the multi-modal information in video data for the text-video retrieval task, we leverage multiple experts [74, 66, 26] to encode raw videos. Specifically, given an input video, we leverage N experts $\{\boldsymbol{E}^1, \boldsymbol{E}^2, \ldots, \boldsymbol{E}^N\}$ to extract multi-modal features. Here $\boldsymbol{E}^n$ reprents the n-th expert. Each expert is pretrained on a particular task to acquire specific knowledge on the corresponding modal. Our goal is to achieve both local and global alignment for text-video retrieval, so we extract features from each temporal segment. For each expert, we obtain a set of segment-level video representations, *i.e.*, $\{\boldsymbol{E}^n(\boldsymbol{x}_1), \boldsymbol{E}^n(\boldsymbol{x}_2), \ldots, \boldsymbol{E}^n(\boldsymbol{x}_T)\}$. Here T is the number of segments, and $\boldsymbol{x}_t$ is the t-th segment from a video. We leverage the following two operations to

further process the segment-level multi-expert features for subsequent global-local alignment.

First, we introduce to generate **global expert features for global alignment**. We aim to perform temporal aggregation for each expert to generate global expert features. There are a few existing temporal aggregation operations to obtain a global vector, *e.g.*, temporal convolution networks [58], Transformers [95] and NetVLAD [2]. For simplicity, we leverage a max-pooling operation without additional parameters. This simple operation works well in our experiments. The output feature of each temporal aggregation module is then projected to the same dimension and enhanced by a self-gating mechanism [74]. Consequently, we obtain a set of global expert feature $\{\boldsymbol{F}_1^{video}, \boldsymbol{F}_2^{video}, \ldots, \boldsymbol{F}_N^{video}\}$, where N is the number experts.

Second, we use a one-layer transformer to **fuse multi-expert features for local alignment**. We first employ a fully-connected layer for each expert to project different expert features to a C-dimensional embedding space. We then combine the features from all experts to generate the local features $\boldsymbol{Z}^{video} = \{\boldsymbol{z}_1^{video}, \boldsymbol{z}_2^{video}, \ldots, \boldsymbol{z}_M^{video}\}$, where M is the number of features from all experts. We further explore the relations among the multi-modal features with self-attention mechanism. This design is similar to [26] but has two differences: (1) We only use a one-layer transformer encoder [95] instead of the multi-layer transformers with pre-aggregation and position encoding as in [26]. Thus, our module introduces fewer parameters and is more computationally efficient; (2) We aim to maintain the locality of the input features while MMT generates aggregated expert features for the subsequent text-to-video matching. The output feature $\boldsymbol{Z}^{video}$ of this process has the same length as the input features.

6.2.3 Text Representation

The BERT model [16] has shown great generalization capabilities in language feature encoding. We leverage a pre-trained BERT model to fairly compared to [26]. The BERT model extracts the contextual word embeddings for each text input. The input sentences are tokenized and padded to be a fixed-length sequence. The fixed-length sequence is the input to the BERT model. We add special token like "[CLS]" and "[SEP]" to indicate the start and the end of the sentence. The features can be computed as $\boldsymbol{Z}^{text} = \Phi^{BERT}(S)$, where Φ^{BERT} is the BERT model, S is the input tokens. $\boldsymbol{Z}^{text} = \{\boldsymbol{z}_1^{text}, \boldsymbol{z}_2^{text}, \ldots, \boldsymbol{z}_B^{text}\}$, where B is the sequence length. The BERT model Φ^{BERT} is optimized with the other modules in our framework in an end-to-end manner. It provides powerful text modeling capacity. Different from video encoding, the global feature for text are extracted jointly with local representations for the subsequent T2VLAD module.

6.2.4 Local Alignment

After the aforementioned text encoding and video encoding, we obtain B local contextual word embeddings $\boldsymbol{Z}^{text}$ and M video local features $\boldsymbol{Z}^{video}$ for each input text-video pair. These features contain abundant information about the input sentences and videos. However, the direct comparisons between the two types of features is not feasible because they are not well-aligned. Moreover, the local video features $\boldsymbol{Z}^{video}$ are from different modalities. The domain gaps increase the difficulties of the local alignment. Intuitively, if we select and aggregate the local text features and video features on the same topic and then compare their similarity, the measurement would become more precise. Motivated by this, we propose a Text-to-Video VLAD (T2VLAD) to cluster the local features from multiple modalities with shared centers. These centers provide shared semantic topics which can bridge the gaps among different modalities. Inspired by [2], these centers can be learned jointly

with the whole network and the feature clustering can be performed on-the-fly.

Specifically, we learn $K+1$ C-dimensional shared cluster centers $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K, \boldsymbol{c}_{K+1}\}$. Here the K centers are for local alignment and the additional center is for background information removal. The design of background center shares the same spirit of [121] to discard noise information. We follow [2] to calculate the similarities between each local feature and the cluster centers using dot-product. This step computes assignments on the corresponding clusters. We start with the encoding of video features. Given a local video feature $\boldsymbol{z}_i^{video}$, its assignment to j-th cluster can be generated as follows,

$$a_{i,j} = \frac{exp(\boldsymbol{z}_i^{video}\boldsymbol{c}_j^{\mathsf{T}} + b_j)}{\sum_{k=1}^{K+1} exp(\boldsymbol{z}_i^{video}\boldsymbol{c}_k^{\mathsf{T}} + b_k)}, \tag{6.1}$$

where b_j is a learnable bias term. In practice, one can replace the bias term with a batch normalization layer [45] which normalizes and shifts the activation by two built-in learnable parameters. Then the aggregated residual feature on each centers can be obtained,

$$\boldsymbol{g}_j^{video} = \text{normalize}(\sum_{i=1}^{M} a_{i,j}(\boldsymbol{z}_i^{video} - \boldsymbol{c}_j')), \tag{6.2}$$

where the $\boldsymbol{c}_j'$ is trainable weights that have the same size as $\boldsymbol{c}_j$, and "normalize" indicates a ℓ_2-normalization operation. The design of introducing two centers for each cluster has been proposed in [2] to increase the adaptation capability of the NetVLAD layer. We obtain a set of aggregated video feature $\boldsymbol{G}^{video} = \{\boldsymbol{g}_1^{video}, \boldsymbol{g}_2^{video}, \ldots, \boldsymbol{g}_K^{video}\}$. Each feature in $\boldsymbol{G}^{video}$ is the aligned local feature for the video. Note that the aggregated feature on the background center is abandoned and not involved in the following similarity measurement.

The aggregated text features can be calculated in the same way using the shared cluster centers.

$$\boldsymbol{g}_j^{text} = \text{normalize}(\sum_{i=1}^{B} \frac{exp(\boldsymbol{z}_i^{text}\boldsymbol{c}_j^{\mathsf{T}} + b_j)}{\sum_{k=1}^{K+1} exp(\boldsymbol{z}_i^{text}\boldsymbol{c}_k^{\mathsf{T}} + b_k)}(\boldsymbol{z}_i^{text} - \boldsymbol{c}_j')), \tag{6.3}$$

where $\boldsymbol{z}_i^{text}$ is the local word embedding in $\boldsymbol{Z}^{text}$, We can obtain the final local feature $\boldsymbol{G}^{text} = \{\boldsymbol{g}_1^{text}, \boldsymbol{g}_2^{text}, \ldots, \boldsymbol{g}_K^{text}\}$ for the text sequence. Since the local feature assignment and aggregation for video and text share the same centers, the final features $\boldsymbol{G}^{video}$ and $\boldsymbol{G}^{text}$ can be aligned effectively. We utilize cosine distance to measure the local similarity between the final video and text features $s_{local} = dist(\boldsymbol{G}^{video}, \boldsymbol{G}^{text})$.

6.2.5 Global Alignment

We introduce global alignment for two reasons. First, the global feature for text-video pairs is more comprehensive and complementary to local features. Second, the elaborate local alignment with trainable centers can be difficult to be optimized when lacking auxiliary supervision, especially the video features consist of multi-modal information.

Therefore, we alleviate the optimization difficulty in global alignment by aggregating and transforming the video feature from each expert independently. Meanwhile, we utilize the concatenation of local text features $\boldsymbol{G}^{text}$ to generate the expert-specific global text representations $\{\boldsymbol{F}_1^{text}, \boldsymbol{F}_1^{text}, \ldots, \boldsymbol{F}_N^{text}\}$. And each feature is then used to compute the similarity with corresponding video expert feature. Following [74], we compute the global text-video similarity as a weighted sum of cosine distances between each global video expert feature and corresponding text feature. Formally, the global similarity is calculated as follows,

$$s_{global} = \sum_{i=1}^{N} w_i * dist(\boldsymbol{F}_i^{text}, \boldsymbol{F}_i^{video}), \tag{6.4}$$

where w_i is the weight for the i-th expert. The weights are generated from the text representation $\boldsymbol{G}^{text}$ by a linear projection with a softmax normalization.

We utilize the text-video similarity $s = \frac{1}{2}(s_{global} + s_{local})$ to obtain a simple bi-directional max-margin ranking loss on both text-to-video and video-to-text retrieval tasks, following [74, 26]. We refer the reader to [74, 26] for detailed descriptions.

6.3 Experiments

6.3.1 Experimental Details

Dataset. We experiment with MSRVTT [112], video-text datasets. The **MSRVTT** dataset contains 10,000 videos. These videos are collected from YouTube using 257 queries from a commercial video search engine. We evaluate the performance on three splits. For the "1k-A" split, the train and test are splitted as introduced in [116]. The "1k-B" split is obtained following [74]. Both splits use 9,000 videos for training and the remaining 1,000 videos are used for testing. The **ActivityNet Captions** dataset [54] consists of 20,000 videos. Each video is densely annotated with multiple sentence descriptions. The **LSMDC** dataset [82] consists of 118,081 short video clips. The videos are extracted from 202 long movies.

Evaluation Metrics. We report the results with the standard video retrieval metrics, *i.e.*, Rank K (R@K, higher is better), Median Rank (MdR, lower is better). We report R@1, R@5, and R@10 following [8, 66].

Multi-Expert Features. We use the features provided by [26] in our experiments. There features are: Motion features from S3D [110] trained on the Kinetics dataset. Audio features from VGGish model [39] trained on YT8M. Scene embeddings from DenseNet-161 [43] trained on the Places365 dataset [123]. We refer the readers to [26] for more descriptions of OCR, Face, Speech and Appearance features. For MSRVTT, we also leverage optical flow features released by [26]. We do not use Speech features on LSMDC due to feature missing from the released features [26].

Implantation Details. We implement our T2VLAD by PyTorch. We train the models from scratch and no additional data is used. The margin in the ranking loss is set to 0.02 for all datasets. Following [66], we leverage Ranger optimizer

with a weight decay 0.0001. We initialize the learning rate at 0.0001, and decay by a multiplicative factor 0.9 every 5 epochs. The batch size of the video-text pairs is set to 64. For text encoding, we use the pretrained BERT model "BERT-base-uncased" and fine-tune it with our framework in an end-to-end manner. For video expert encoding, we leverage the pre-extracted expert features provided by [26]. We use all the 8 experts for the MSRVTT dataset and 6 experts (rgb, audio, ocr, scene, flow and action) for the LSMDC dataset. For ActivityNet Captions, we only use the motion and audio experts. The self-attention module used for local video features is implemented by 1 layer multi-head attention with 4 heads, a dropout probability of 0.1 and the hidden size of 768. The dimension for the common space of both global alignment and local alignment is also set to 768. We set the center size of our T2VLAD to 9 for short video retrieval dataset (MSRVTT and LSMDC) and 16 for long video retrieval dataset.

6.3.2 Comparison to State-of-the-art

MSRVTT. The results on MSRVTT are shown in Table 6.1. We consistently improve the state-of-the-art on text-to-video retrieval and video-to-text retrieval across all three splits. MMT [26] is recently proposed to perform text-video retrieval using multi-modal transformers. MMT achieved the best performance in the compared methods. Notably, during text-to-video retrieval, we outperform MMT [26] with 5.8% gain on the R@1 metric on the 1k-B split (20.3% vs. 26.1%). A 5.6% improvement on R@1 (1k-B split) is also obtained compared to MMT [26] (21.1% vs. 26.7%). These results demonstrate the benefits of our T2VLAD in cross-modal retrieval tasks. Notably, we obtain consistent improvements over "MMT +HT pretrain" [26] on the 1k-A split. Note that "MMT + HT pretrain" is pre-trained on a large-scale instructional video dataset, *i.e.*, HowTo100M, containing more than one million videos with machine generated descriptions. Pre-training on

Method	Split	Text → Video				Video → Text			
		R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑	R@10↑	MdR↓
JSFusion [116]	1k-A	10.2	31.2	43.2	13	-	-	-	-
HT [75]	1k-A	14.9	40.2	52.8	9	-	-	-	-
CE [66]	1k-A	20.9	48.8	62.4	6	20.6	50.3	64.0	5.3
MMT [26]	1k-A	24.6	54.0	67.1	4.0	24.4	56.0	67.8	4.0
MMT + HT pretrain [26]	1k-A	26.6	57.1	69.6	4.0	27.0	57.5	69.7	3.7
Our T2VLAD	1k-A	**29.5**	**59.0**	**70.1**	**4**	**31.8**	**60.0**	**71.1**	**3**
MEE [74]	1k-B	13.6	37.9	51.0	10.0	-	-	-	-
JPose [102]	1k-B	14.3	38.1	53.0	9	16.4	41.3	54.4	8.7
MEE-COCO [74]	1k-B	14.2	39.2	53.8	9.0	-	-	-	-
CE [66]	1k-B	18.2	46.0	60.7	7.0	18.0	46.0	60.3	6.5
MMT [26]	1k-B	20.3	49.1	63.9	6.0	21.1	49.4	63.2	6.0
Our T2VLAD	1k-B	**26.1**	**54.7**	**68.1**	**4**	**26.7**	**56.1**	**70.4**	**4**
VSE [76]	Full	5.0	16.4	24.6	47	7.7	20.3	31.2	28
VSE++ [76]	Full	5.7	17.1	24.8	65	10.2	25.4	35.1	25
Mithun et al. [76]	Full	7.0	20.9	29.7	38	12.5	32.1	42.4	16
W2VV [18]	Full	6.1	18.7	27.5	45	11.8	28.9	39.1	21
Dual Enc. [19]	Full	7.7	22.0	31.8	32	13.0	30.8	43.3	15
HGR [8]	Full	9.2	26.2	36.5	24	15.0	36.7	48.8	11
E2E [72]	Full	9.9	24.0	32.4	29.5	-	-	-	-
CE [66]	Full	10.0	29.0	41.2	16	15.6	40.9	55.2	8.3
Our T2VLAD	Full	**12.7**	**34.8**	**47.1**	**12**	**20.7**	**48.9**	**62.1**	**6**

Table 6.1 : The comparison with the state-of-the-art methods on the MSRVTT [112] dataset.

HowTo100M significantly improves the performance of MMT across all evaluation metrics. T2VLAD does not leverage additional training videos, but we outperform "MMT +HT pretrain" on split 1k-A with a clear margin across all metrics. For instance, on text-to-video retrieval, T2VLAD outperforms "MMT +HT pretrain" by 2.9% at R@1. These results demonstrate that the benefit of the global-local

Method	Text → Video				Video → Text			
	R@1 ↑	R@5 ↑	R@50 ↑	MdR ↓	R@1 ↑	R@5 ↑	R@50 ↑	MdR ↓
FSE [118]	18.2	44.8	89.1	7	16.7	43.1	88.4	7
CE [66]	18.2	47.7	91.4	6	17.7	46.6	90.9	6
HSE [118]	20.5	49.3	-	-	18.7	48.1	-	-
MMT [26]	22.7	54.2	93.2	5.0	22.9	54.8	93.1	4.3
Ours	**23.7**	**55.5**	**93.5**	**4**	**24.1**	**56.6**	**94.1**	**4**

Table 6.2 : The comparisons with the state-of-the-art methods on the ActivityNet Captions dataset.

alignment using T2VLAD.

ActivityNet Captions. The results on ActivityNet Captions are shown in Table 6.2. The compared baselines include HSE [8], CE [66], HSE [118], and MMT [26]. HSE leverages a hierarchical sequence embedding and MMT incorporates multi-layer transformers for strong video feature learning. We consistently improve MMT over all benchmark metrics.

LSMDC. The LSMDC data is collected from movies. We report on LSMDC to show that our T2VLAD is capable of dealing with different videos from different domains. The results are shown in Table 6.3. We observe consistent improvements over MMT. For instance, we achieve 2.1% improvements on R@1 for video-to-text retrieval.

6.3.3 Ablation Study

The effectiveness of the global-local alignment. In Table 6.4, we shows the results of only using the single alignment of our model. To implement the model

Method	Text → Video				Video → Text			
	R@1 ↑	R@5 ↑	R@50 ↑	MdR ↓	R@1 ↑	R@5 ↑	R@50 ↑	MdR ↓
CT-SAN [117]	5.1	16.3	25.2	46	-	-	-	-
JSFusion [116]	9.1	21.2	34.1	36	-	-	-	-
CCA [52]	7.5	21.7	31.0	33	-	-	-	-
MEE [74]	9.3	25.1	33.4	27	-	-	-	-
MEE-COCO [74]	10.1	25.6	34.6	27	-	-	-	-
CE [66]	11.2	26.9	34.8	25.3	-	-	-	-
MMT [26]	13.2	29.2	38.8	21.0	12.1	29.3	37.9	22.5
Ours	**14.3**	**32.4**	**42.2**	**16**	**14.2**	**33.5**	**41.7**	**17**

Table 6.3 : The comparison with the state-of-the-art methods on the LSMDC dataset.

without local alignment, we follow [26] to utilize the "[CLS]" output of BERT model as the global text representation and remove the local branch. We find that the performance drops a lot only using the global alignment for both text-to-video and video-to-text retrieval. It demonstrates the effectiveness of the local alignment method in the cross-modal retrieval task. When we remove the local alignment branch and only train the global alignment, the test performance drops a lot compared to the results of our full model. This proves our local alignment is crucial for cross-modal retrieval task. When we remove the global alignment and only train the local alignment, the loss can not converge. It demonstrates the importance of global alignment for providing additional supervision to the precise local alignment. We show the results of removing the global alignment only at test time, *i.e.*, "Ours w/o Global Alignment" in Table 6.4. Compared to the full model, the results drop by a large margin. Our results prove that the global feature is complementary to the local information, even though local alignment can enable

Method	Text → Video				Video → Text			
	R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑	R@10↑	MdR↓
Ours w/o Global Alignment	24.3	51.5	63.4	5	26.6	52.9	62.6	5
Ours w/o Local Alignment	22.2	49.9	64.6	6	24.0	51.7	65.6	5
Full model	**29.5**	**59.0**	**70.1**	**4**	**31.8**	**60.0**	**71.1**	**3**

Table 6.4 : The ablation studies on the MSRVTT [112] dataset to investigate the effectiveness of global-local alignment.

Method	Text → Video				Video → Text			
	R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑	R@10↑	MdR↓
Ours w/ only text VLAD	27.4	57.3	68.2	4	27.5	57.4	69.7	4
Ours w/ two separate VLAD	28.6	58.1	70.4	4	30.4	60.7	72.1	3
Ours w/ two shared VLAD	**29.5**	**59.0**	70.1	**4**	**31.8**	60.0	71.1	**3**

Table 6.5 : The ablation studies on the MSRVTT [112] dataset to investigate the effectiveness of the VLAD encoding.

precise comparisons.

The effectiveness of collaborative VLAD. In "Ours w/ only text VLAD", we replace the shared NetVLAD layer for local video feature encoding with a max-pooling operation and then project the feature to the same dimension of text local features. This model achieves lower performance than our T2VLAD, showing the importance of joint VLAD encoding. In "Ours w/ two separate VLAD", we do not perform center sharing between text feature encoding and video feature encoding. The VLAD centers are learned separately. The results show that our strategy of sharing centers outperform "Ours w/ two separate VLAD" in both text-to-video retrieval and video-to-text retrieval. This demonstrate that our center sharing idea is beneficial to reduce the semantic gap between text and video data.

Method	Text → Video				Video → Text			
	R@1↑	R@5↑	R@10↑	MdR↓	R@1↑	R@5↑	R@10↑	MdR↓
Ours w/o Visual Att	26.8	58.3	70.7	4	31.3	60.5	71.2	3
Full model	**29.5**	**59.0**	70.1	**4**	**31.8**	60.0	71.1	**3**

Table 6.6 : The ablation studies on the MSRVTT [112] dataset to investigate the effectiveness of the Visual Attention.

The effectiveness of the visual attention We evaluate the model without the self-attention layer in the local visual branch. As shown in Table 6.6, compared to the results of our full model, the rank-1 accuracy drops 2.7% and the other metrics are comparable. The self-attention layer encourages the local features to integrate the information from other modalities and temporal locations. But it is not the core module in our framework. Without this layer, our method also achieves the state-of-the-art.

6.3.4 Qualitative Results

Visualization of the assignments. The text local features an the video local features are assigned to set of shared centers in our T2VLAD. We expect the aggregated text feature and video feature on the same center share the similar topic. In Fig. 6.3, we illustrate the text assignments and video appearance feature assignment on three centers. The video is ranked first in the text retrieved results. The thickness of the connections reflects the value of the text assignment. We show the video frames corresponding to the appearance features that are assigned to the certain center. As shown in Fig. 6.3, the text feature with highest assignment on the Center 1 is the feature of "guy". All the frames that have been assigned to Center 1 also contains the appearance information of "guy". The text with highest assignment on Center 2 is "something", and the only frame assigned to the center is about the

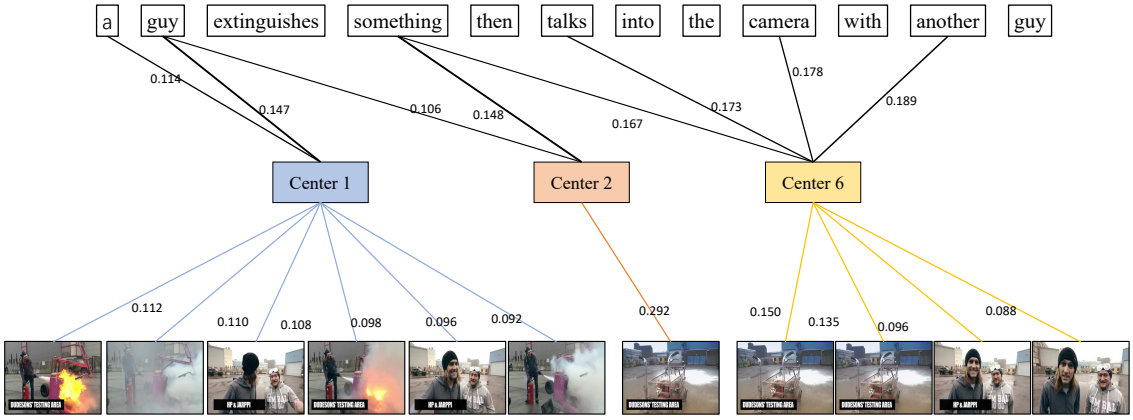

Figure 6.3 : Visualization of the assignment weights. We take Video 7060 in the MSRVTT 1K-A test set as an example. We plot the text assignments to the three centers as black lines. The thickness of the line indicates the relative value on the same center. The numbers next to the line are the assignment values. We only illustrate the top text assignments for better visualization. The Top-10 frames (the padding features have been removed.) correspond to the appearance features assigned to the centers are shown at the bottom.

"something" in the video. On Center 6, the text "something", "talks", "camera" and "another" all have high assignments. And the frames assigned to the center contain the content of this semantic meaning. Interestingly, the most salient word "extinguishes" in human view always has a low assignment value on all centers. We think this is because the limited training data is not enough to enable the understanding on complex and low-frequency word. The assignments visualization verifies that our T2VLAD can achieve adequate local alignment for text-to-video retrieval.

Visualization of the text-to-video results. We show two examples of the videos that retrieved by our method and the model without local alignment branch. As shown in Fig. 6.4, the two query sentences consist of multiple semantic cues. Our T2VLAD successfully retrieves the ground-truth video while the model with-

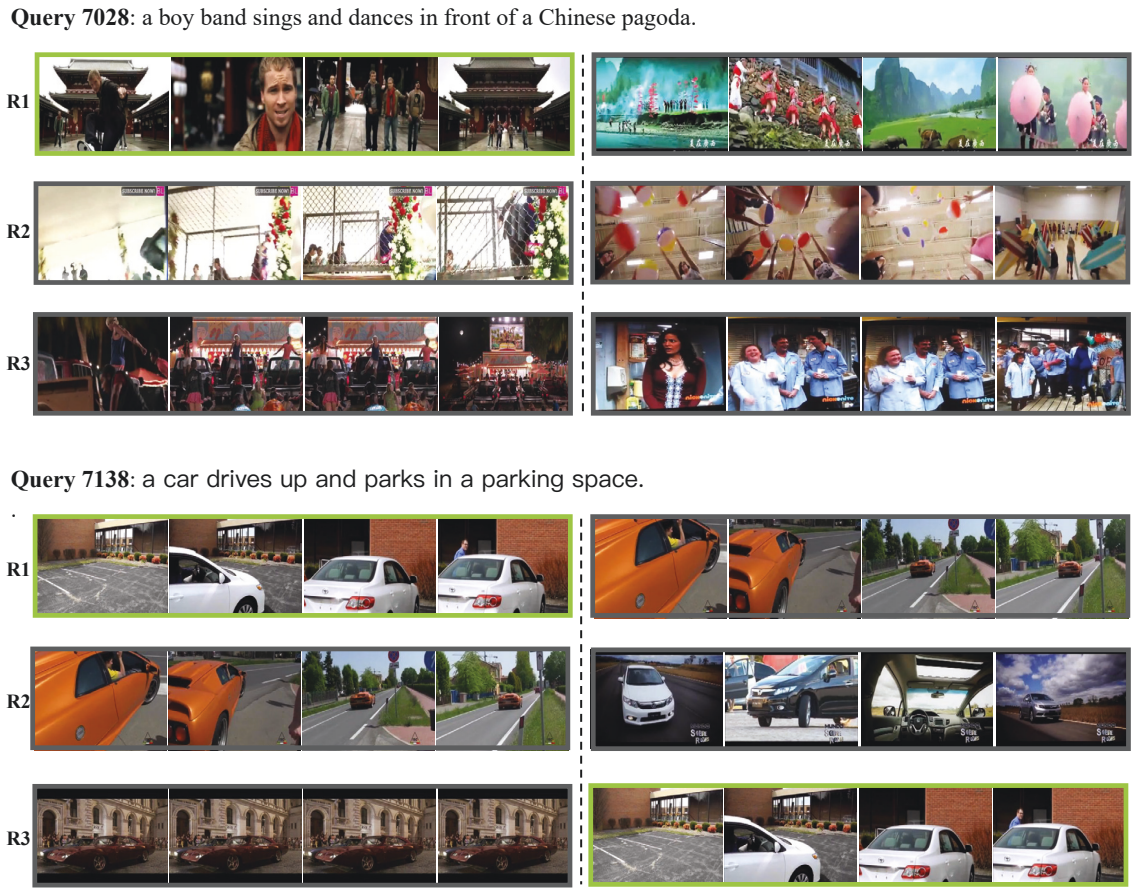

Figure 6.4 : The text-video retrieval results on the MSRVTT 1K-A test set. The left are the videos ranked by our T2VLAD, and the right are the results from the model only with global alignment.

out local alignment returns several videos that are somewhat relevant to the query sentence but are not precise. In the second example, our T2VLAD achieves a better alignment between the text and videos on the local semantic cue "parks".

6.4 Summary

In this Chapter, we introduce an end-to-end text-video sequence alignment method without complex graph operations and multi-layer transformers. We show that local semantic alignment between texts and videos is critical for high-performance retrieval systems. We achieve the goal of local alignment based on NetVLAD and

introduce T2VLAD for collaborative text-video encoding. Our results on three standard text-to-video and video-to-text benchmarks clearly demonstrate the effectiveness of our method. The visualization results also validate our motivation of joint semantic topic learning. In the future, more efforts could be paid in obtaining better global video features.

Chapter 7

Conclusion and Future Directions

In this thesis, we investigated collaborative dual-stream modeling for video understanding. Compared to image analysis, video understanding require the system to model the complex temporal information and reduce the redundant computation. Previous single-stream modeling methods with only one backbone or single modality limit the communications among different modalities or CNN models. Compared to these existing works, the proposed collaborative dual-stream methods improve the reasoning capability of the video understanding system, which leads to better performance in complex application scenarios. We contributed to three tasks, including egocentric action recognition, text-video retrieval, and efficient video recognition. However, much work remains to be done in the future for video understanding.

For egocentric action recognition, we proposed two novel frameworks in Chapter 3 and Chapter 4 to enable the interaction between the motion feature and the object feature. The communications between these two streams help the model to locate and recognize the complex active objects in egocentric videos. The proposed methods only take the video clip in the trimmed video segments as input. So the long-term temporal information in the whole video has been overlooked in our frameworks. We can consider the context information in the entire video and leverage the high-level semantic relationship among the sequential segments to improve the accuracy of action recognition. Another potential way is to make full use of the multi-modal information in egocentric videos such as audio and human gaze.

For efficient video recognition, we proposed a differentiable parallel sampling

network to select the most salient frames in the input video. During training, the interaction between the lightweight sampler and the heavyweight recognition model is enabled by a weighted sum operation. The number of the sampled frames is pre-defined in our pipeline. It may lead to some failure cases when the input video is too short. And for very complex videos, the pre-defined number of frames may not be adequate for accurate recognition. In the future, we can enable the model to sample an appropriate number of frames based on the input videos. Besides, we can embed the features from the two models into a shared space and learn the salient scores by more effective schemes like metric learning.

For text-video retrieval, we developed a global-local sequence alignment framework in Chapter 6 to better measure the text-video similarities in the joint space. The model is optimized by a max-margin ranking loss calculated in a mini-batch. In the current framework, we only leverage the text-video pairs to construct the triplet for optimization. One future direction is looking for latent text-video supervision signals by more precise text-text association.

Bibliography

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 5297–5307.

[3] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori, "Object level visual reasoning in videos," in *Eur. Conf. Comput. Vis.*, 2018.

[4] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient video classification using fewer frames," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[5] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[6] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[8] S. Chen, Y. Zhao, Q. Jin, and Q. Wu, "Fine-grained video-text retrieval with hierarchical graph reasoning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 638–10 647.

[9] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2018.

[10] Z. Chen, Y. Li, S. Bengio, and S. Si, "You look twice: Gaternet for dynamic filter selection in cnns," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[11] D. Damen, H. Doughty, G. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "The epic-kitchens dataset: Collection, challenges and baselines," *IEEE IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[12] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *Eur. Conf. Comput. Vis.*, 2018.

[13] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Rescaling egocentric vision," *arXiv preprint arXiv:2006.13256*, 2020.

[14] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. W. Mayol-Cuevas, "You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video." in *BMVC*, vol. 2, 2014, p. 3.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[18] J. Dong, X. Li, and C. G. Snoek, "Predicting visual features from text for image and video caption retrieval," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3377–3388, 2018.

[19] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang, "Dual encoding for zero-example video retrieval," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 9346–9355.

[20] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," *arXiv preprint arXiv:1707.05612*, 2017.

[21] H. Fan, Z. Xu, L. Zhu, C. Yan, J. Ge, and Y. Yang, "Watching a small portion could be as good as watching all: Towards efficient video classification," in *IJCAI*, 2018.

[22] H.-S. Fang, J. Cao, Y.-W. Tai, and C. Lu, "Pairwise body-part attention for recognizing human-object interactions," in *Eur. Conf. Comput. Vis.*, 2018.

[23] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *IEEE Conf. Comput. Vis. Pattern Recog. 2011.* IEEE, 2011, pp. 3281–3288.

[24] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6202–6211.

[25] A. Furnari, S. Battiato, K. Grauman, and G. M. Farinella, "Next-active-object prediction from egocentric videos," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 401–411, 2017.

[26] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, "Multi-modal transformer for video retrieval," in *Eur. Conf. Comput. Vis.*, 2020.

[27] R. Gao, T.-H. Oh, K. Grauman, and L. Torresani, "Listen to look: Action recognition by previewing audio," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[28] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[29] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.

[30] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[31] R. Girshick, "Fast r-cnn," in *Int. Conf. Comput. Vis.*, 2015.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[33] G. Gkioxari, R. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[34] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The" something something" video database for learning and evaluating visual common sense." in *Int. Conf. Comput. Vis.*, 2017.

[35] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[36] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn." *IEEE TPAMI*, 2018.

[37] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn," in *Int. Conf. Comput. Vis.*, 2017.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[39] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *ICASSP*. IEEE, 2017, pp. 131–135.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

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

[42] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[44] Y. Huang, M. Cai, Z. Li, F. Lu, and Y. Sato, "Mutual context network for jointly estimating egocentric gaze and action," *IEEE Transactions on Image Processing*, vol. 29, pp. 7795–7806, 2020.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[46] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2010, pp. 3304–3311.

[47] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[48] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3128–3137.

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

[50] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *Int. Conf. Comput. Vis.*, 2019.

[51] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:1411.2539*, 2014.

[52] B. Klein, G. Lev, G. Sadeh, and L. Wolf, "Associating neural word embeddings with deep image representations using fisher vectors," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 4437–4446.

[53] B. Korbar, D. Tran, and L. Torresani, "Scsampler: Sampling salient clips from video for efficient action recognition," in *Int. Conf. Comput. Vis.*, 2019.

[54] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *Int. Conf. Comput. Vis.*, 2017, pp. 706–715.

[55] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, 2016.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[57] Z. Lan, Y. Zhu, A. G. Hauptmann, and S. Newsam, "Deep local video feature for action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 1–7.

[58] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 156–165.

[59] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Eur. Conf. Comput. Vis.*, 2018.

[60] ——, "In the eye of the beholder: Gaze and actions in first person video," *arXiv preprint arXiv:2006.00626*, 2020.

[61] Y. Li, Z. Ye, and J. M. Rehg, "Delving into egocentric actions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 287–295.

[62] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Int. Conf. Comput. Vis.*, 2019.

[63] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *ICLR*, 2017.

[64] D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre, "Learning what and where to attend," in *International Conference on Learning Representations*, 2019.

[65] M. Liu, S. Tang, Y. Li, and J. Rehg, "Forecasting human object interaction: Joint prediction of motor attention and egocentric activity," in *Eur. Conf. Comput. Vis.*, 2020.

[66] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman, "Use what you have: Video retrieval using representations from collaborative experts," in *BMVC*, 2019.

[67] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[68] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[69] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[70] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Eur. Conf. Comput. Vis.*, 2018.

[71] Y. Meng, C.-C. Lin, R. Panda, P. Sattigeri, L. Karlinsky, A. Oliva, K. Saenko, and R. Feris, "Ar-net: Adaptive frame resolution for efficient action recognition," in *Eur. Conf. Comput. Vis.*, 2020.

[72] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 9879–9889.

[73] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," *arXiv preprint arXiv:1706.06905*, 2017.

[74] ——, "Learning a text-video embedding from incomplete and heterogeneous data," *arXiv preprint arXiv:1804.02516*, 2018.

[75] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Int. Conf. Comput. Vis.*, 2019.

[76] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury, "Learning joint embedding with multimodal cues for cross-modal video-text retrieval," in *ICMR*, 2018, pp. 19–27.

[77] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 4594–4602.

[78] W. Price and D. Damen, "An evaluation of action recognition models on epic-kitchens," *arXiv preprint arXiv:1908.00867*, 2019.

[79] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *Eur. Conf. Comput. Vis.*, 2018.

[80] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Int. Conf. Comput. Vis.*, 2017.

[81] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

[82] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3202–3212.

[83] F. Sener, D. Singhania, and A. Yao, "Temporal aggregate representations for long-range video understanding," in *European Conference on Computer Vision*. Springer, 2020, pp. 154–171.

[84] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and A. Karteek, "Actor and observer: Joint modeling of first and third-person videos," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7396–7404, 2018.

[85] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NeurIPS*, 2014.

[86] S. Singh, C. Arora, and C. Jawahar, "First person action recognition using deep learned descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2620–2628.

[87] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Pose-driven deep convolutional model for person re-identification," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3960–3969.

[88] S. Sudhakaran, S. Escalera, and O. Lanz, "Fbk-hupba submission to the epic-kitchens 2019 action recognition challenge," *arXiv preprint arXiv:1906.08960*, 2019.

[89] ——, "Lsta: Long short-term attention for egocentric action recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[90] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," in *BMVC*, 2018.

[91] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid, "Actor-centric relation network," in *Eur. Conf. Comput. Vis.*, 2018.

[92] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Int. Conf. Comput. Vis.*, 2015.

[93] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *Int. Conf. Comput. Vis.*, 2019.

[94] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[95] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[96] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

[97] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Eur. Conf. Comput. Vis.*, 2016.

[98] X. Wang, Y. Wu, L. Zhu, and Y. Yang, "Symbiotic attention with privileged information for egocentric action recognition," in *AAAI*, 2020.

[99] X. Wang, L. Zhu, Y. Wu, and Y. Yang, "Symbiotic attention for egocentric action recognition with object-centric alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[100] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[101] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Eur. Conf. Comput. Vis.*, 2018.

[102] M. Wray, D. Larlus, G. Csurka, and D. Damen, "Fine-grained action retrieval through multiple parts-of-speech embeddings," in *Int. Conf. Comput. Vis.*, 2019.

[103] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[104] W. Wu, D. He, X. Tan, S. Chen, and S. Wen, "Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition," in *Int. Conf. Comput. Vis.*, 2019.

[105] Y. Wu, L. Zhu, Y. Yan, and Y. Yang, "Dual attention matching for audio-visual event localization," in *Int. Conf. Comput. Vis.*, 2019.

[106] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification," in *ACM Multimedia*, 2016, pp. 791–800.

[107] Z. Wu, C. Xiong, Y.-G. Jiang, and L. S. Davis, "Liteeval: A coarse-to-fine framework for resource efficient video recognition," in *NeurIPS*, 2019.

[108] Z. Wu, C. Xiong, C.-Y. Ma, R. Socher, and L. S. Davis, "Adaframe: Adaptive frame selection for fast video recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[109] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer, "Audiovisual slowfast networks for video recognition," *arXiv preprint arXiv:2001.08740*, 2020.

[110] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Eur. Conf. Comput. Vis.*, 2018.

[111] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *NeurIPS*, 2003, pp. 521–528.

[112] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 5288–5296.

[113] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with

visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[114] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative cnn video representation for event detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[115] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[116] Y. Yu, J. Kim, and G. Kim, "A joint sequence fusion model for video question answering and retrieval," in *Eur. Conf. Comput. Vis.*, 2018, pp. 471–487.

[117] Y. Yu, H. Ko, J. Choi, and G. Kim, "End-to-end concept word detection for video captioning, retrieval, and question answering," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3165–3173.

[118] B. Zhang, H. Hu, and F. Sha, "Cross-modal and hierarchical modeling of video and text," in *Eur. Conf. Comput. Vis.*, 2018, pp. 374–390.

[119] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1224–1244, 2017.

[120] Y.-D. Zheng, Z. Liu, T. Lu, and L. Wang, "Dynamic sampling networks for efficient action recognition in videos," *IEEE TIP*, 2020.

[121] Y. Zhong, R. Arandjelović, and A. Zisserman, "Ghostvlad for set-based face recognition," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 35–50.

[122] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Eur. Conf. Comput. Vis.*, 2018.

[123] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *TPAMI*, vol. 40, no. 6, pp. 1452–1464, 2017.

[124] C. Zhu, X. Tan, F. Zhou, X. Liu, K. Yue, E. Ding, and Y. Ma, "Fine-grained video categorization with redundancy reduction attention," in *Eur. Conf. Comput. Vis.*, 2018.

[125] L. Zhu, Z. Xu, and Y. Yang, "Bidirectional multirate reconstruction for temporal modeling in videos," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[126] Y. Zhu, Z. Lan, S. Newsam, and A. Hauptmann, "Hidden two-stream convolutional networks for action recognition," in *ACCV*, 2018.

[127] Y. Zhu and S. Newsam, "Efficient action detection in untrimmed videos via multi-task learning," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 197–206.