UTS UNIVERSITY OF TECHNOLOGY SYDNEY

# Multimodal Learning and Video Analysis with Deep Neural Networks

**by Yu Wu**

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

Sept 2021

# Certificate of Authorship/Originality

I, Yu Wu, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed
prior to publication.

Sept 5 2021

# ABSTRACT

**Multimodal Learning and Video Analysis with Deep Neural Networks**

by

Yu Wu

Multi-modal perception is essential when we human explore, capture and perceive the real world. As a multi-modal media, video captures informative content in our daily life. Although deep-learning-based networks have proven to be successful in understanding visual images, an intelligent system is expected to perceive the world from the overall understanding of multiple modalities (e.g., vision and audio), and communicate humans with natural language.

This thesis introduces several works on multi-modal perception and video analysis, including audio-visual video understanding, anticipating future actions, and describing unseen visual content using natural languages. For detailed analyzing audio-visual events in videos, I present a double attention corresponding network for synchronized audio-visual events and exploring heterogeneous clues for asynchronous audio-visual video parsing. For anticipating future actions, I propose to generate intermediate future features and optimize the generation via contrastive learning for multiple modality sources. For visual captioning, I design a decoupled novel object captioner to generate generalized captioning sentences for unseen objects.

Dissertation directed by Professor Yi Yang
School of Computer Science

# Acknowledgements

First and foremost I am very grateful to my supervisor, Dr. Yi Yang, for his invaluable guidance and support. Pursuing a PhD degree under his supervision is the most important and lucky decision I have ever made. I have been extremely lucky to have a supervisor who encouraged me to set a high standard for my research and build my research career. Without his continuous and invaluable support, the PhD degree would not have been achievable.

I would like to thank my co-supervisor Dr. Linchao Zhu. He introduced me to the multi-modal perception and video analysis area. He gave many inspirational ideas in my doctoral studies. I want to thank my collaborator Lu Jiang, who contributed a lot in guiding me in my research on the vision and language area.

I would also like to thank all my colleagues and friends in the ReLER group, who have made my study and life a wonderful time.

Lastly I would like to thank my parents Buhai Wu and Qingfang Liu for their love and support over so many years. I want to thank my wife Yutian Lin for her love and encouragement.

Yu Wu

Sydney, Australia, 2021.

# List of Publications

### Journal Papers

J-1. **Y. Wu**, L. Zhu, X. Wang, Y. Yang and F. Wu, "Learning to Anticipate Egocentric Actions by Imagination," in *IEEE Transactions on Image Processing*, vol. 30, pp. 1143-1152, 2021,

J-2. R. Quan, **Y. Wu**, X. Yu and Y. Yang, "Progressive Transfer Learning for Face Anti-Spoofing," in *IEEE Transactions on Image Processing*, vol. 30, pp. 3946-3955, 2021.

J-3. J. Miao, **Y. Wu** and Y. Yang, "Identifying Visible Parts via Pose Estimation for Occluded Person Re-Identification," in *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

J-4. M. Fan, **Y. Wu**, X. Yu and Y. Yang, "Learning with Noisy Labels via Self-Reweighting from Class Centroids," in *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

J-5. R. Quan, L. Zhu, **Y. Wu** and Y. Yang, "Holistic LSTM for Pedestrian Trajectory Prediction," in *IEEE Transactions on Image Processing*, vol. 30, pp. 3229-3239, 2021.

J-6. 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*, 2021.

J-7. **Y. Wu**, L. Jiang and Y. Yang, "Revisiting EmbodiedQA: A Simple Baseline and Beyond," in *IEEE Transactions on Image Processing*, vol. 29, pp. 3984-3992, 2020.

J-8. Y. Lin, **Y. Wu**, C. Yan, M. Xu and Y. Yang, "Unsupervised Person Re-identification via Cross-Camera Similarity Exploration," in *IEEE Transactions*

*on Image Processing*, vol. 29, pp. 5481-5490, 2020.

J-9. Q. Feng, **Y. Wu**, H. Fan, C. Yan, M. Xu and Y. Yang, "Cascaded Revision Network for Novel Object Captioning," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3413-3421, 2020.

J-10. **Y. Wu**, Y. Lin, X. Dong, Y. Yan, W. Bian and Y. Yang, "Progressive Learning for Person Re-Identification With One Example," in *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2872-2881, 2019.

J-11. Y. Lin, L. Zheng, Z. Zheng, **Y. Wu**, Z. Hu, C. Yan and Y. Yang. "Improving person re-identification by attribute and identity learning", in *Pattern Recognition*, vol. 95, pp. 151-161, 2019.

**Conference Papers**

C-1. **Y. Wu** and Y. Yang, "Exploring Heterogeneous Clues for Weakly-Supervised Audio-Visual Video Parsing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

C-2. J. Miao, Y. Wei, **Y. Wu**, C. Liang, G. Li and Y. Yang, "VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild," in *IEEE Conference on Computer Vision and Pat- tern Recognition (CVPR)*, 2021.

C-3. Y. Zhu, **Y. Wu**, H. Latapie, Y. Yang and Y. Yan., "Learning Audio-Visual Correlations from Variational Cross-Modal Generation," in *The international Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2021.

C-4. Y. Zhu, **Y. Wu**, Y. Yang and Y. Yan, "Describing Unseen Videos via Multi-Modal Cooperative Dialog Agents," in *European Conference on Computer Vision (ECCV)*, pp. 153-169, 2020.

C-5. Y. Lin, L. Xie, **Y. Wu**, C. Yan and Q. Tian, "Unsupervised Person Re-identification via Softened Similarity Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3390-3399, 2020.

C-6. Z. Yang, L. Zhu, **Y. Wu** and Y. Yang, "Gated Channel Transformation for Visual Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11794-11803, 2020.

C-7. M. Qi, J. Qin, **Y. Wu** and Y. Yang, "Imitative Non-Autoregressive Modeling for Trajectory Forecasting and Imputation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12736-12745, 2020.

C-8. X. Wang, **Y. Wu**, L. Zhu, Y.Yang, "Symbiotic Attention with Privileged Information for Egocentric Action Recognition," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 12249-12256, 2020.

C-9. **Y. Wu**, L. Zhu, Y. Yan and Y. Yang, "Dual Attention Matching for Audio-Visual Event Localization," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 6292-6300, 2019.

C-10. J. Miao, **Y. Wu**, P. Liu, Y. Ding and Y. Yang. *Pose-Guided Feature Alignment for Occluded Person Re-identification*, in *IEEE International Conference on Computer Vision (ICCV)*, pp. 542-551, 2019.

C-11. R. Quan, X. Dong, **Y. Wu**, L. Zhu and Y. Yang. *Auto-ReID: Searching for a Part-aware ConvNet for Person Re-Identification*, in *IEEE International Conference on Computer Vision (ICCV)*, pp. 3750-3759, 2019.

C-12. **Y. Wu**, L. Zhu, L. Jiang, and Y. Yang. *Decoupled Novel Object Captioner*, in *ACM International Conference on Multimedia (MM)*, pp. 1029-1037, 2018.

C-13. **Y. Wu**, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang. *Exploit the Unknown Gradually: One-Shot Video-Based Person Re-Identification by Stepwise Learning*, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5177-5186, 2018.

# Contents

# List of Figures

# Chapter 1

# Introduction

Video conveys informative content in our daily life, which contains multiple modalities such as vision, audio, and natural languages. The analysis of videos would help understand human behaviors and events. The goal of multimodal perception and video analysis is to recognize the ongoing events or anticipate future actions, which is challenging due to the wide variations of video content.

How to associate the concepts among different modalities is a great challenge in the multi-modal learning field. Different from single modality tasks where objects/instances of a class usually share similar patterns, the same semantic instance would have different patterns in different modalities. In addition, most multi-modal data only have a weak supervision, *i.e.*, whether the two modalities are synchronized or generally align. Thus it is not easy for models to learn the essential relation from the multi-modal data.

In this theses, I study the multi-modal learning and video analysis task, and improve the multi-modal association by leveraging self-supervised temporal and modal relation. I first study the relation of audio and vision for event videos, where I found the encouraging the cross-modal correlation on the temporal axis is the key to solve the cross-modal audio-visual video recognition task. Then I extend the work from multi-modal recognition to the multi-modal anticipation, which is to recognize the event/action before it happens. Similar to the previous recognition models, I found the temporal contrastive learning still improves the overall performances if we can fill in the temporal gap by imaging the missing frames. Lastly, I study the multi-modal

generation task, *i.e.*, the novel object captioning. Different from previous recognition and anticipation tasks, the captioning focuses on generating cross-modal knowledge based on the vision input. But similarly I found the cross-modal attention matching on objects and words is still very useful for such cross-modal generation task.

The contributions of the thesis are listed as follows.

- I propose to introducing self-supervision to improve the cross-modal association in the multi-modal learning field. Some useful self-supervision signals could be temporal alignment and modality co-occurrence. I have conducted experiments on several multi-modal learning tasks such as audio-visual event localization, audio-visual video parsing, and multi-modal video action anticipation. These clearly show the benefits of introducing self-supervision signals.

- I propose to leveraging object-level supervision as the guidance for multi-modal learning. Object-level supervision is more detailed compared to the overall alignment supervision. Leveraging such object-level supervision could benefits the cross-modal association and lead to more meaningful and explainable cross-modal models.

The following introduces the background of audio-visual video understanding, action forecasting based on multiple modalities, and describing novel visual content via natural languages.

## 1.1 Audio-visual Video Understanding

Vision and sound are the most informative sensory streams that we humans to perceive the world. As a good record of our daily life, the video contains these two raw modalities. To better understand human behaviors in videos, it is essential to perceive and model both audio and visual modalities since both of them contain important clues about the ongoing event in videos.

In this thesis, I focus on audio-visual video understanding, which is designed

to find the temporal positions and categories of events in untrimmed videos. The model needs to detect, localize, and recognize events in videos using both audio stream and visual stream. Specifically, I study two kinds of audio-visual events localization tasks, *i.e.*, synchronous audio-visual event localization and asynchronous audio-visual video parsing. The first one is designed to find both audible and visible events at the same time (synchronous), while the second one is to find out all events in each audio track and visual track, respectively. The second one (audio-visual video parsing) is more general and more challenging than the first one (audio-visual event localization).

In Chapter 3, I study the synchronous audio-visual event localization task. Existing works [95, 58] firstly process the input video and cut it into small clips. After that, they combine vision and sound features at the clip level. However, these clips are very short, making the vision and sound features not matched well (misalignment). The previous methods that concatenate audio and vision features at the clip level might be very fragile to such incremental misplacement. Differently, I propose the Double Attention Corresponding (DAC) method. The core idea is that DAC looks at the whole video to obtain the global knowledge, and then attain local event clues by the global-to-local cross-modal comparison. The motivation is that we believe we humans should see the global video to access the overall event information, and look into each clip in detail with the guidance of global concept. To introduce the interaction between auditory and visual features, I further leverage the cross-modal attention mechanism, *i.e.*, taking the overall video feature of one modality to query the clip feature in the other modality. Experiments validate that the proposed DAC significantly beats the state-of-the-art methods.

In Chapter 4, I further study the more general audio-visual understanding task, asynchronous audio-visual video parsing, which aims to parse a video into temporal event segments and predict the audible or visible event categories. The task is chal-

lenging since there only exist video-level event labels for training, without indicating the temporal boundaries and modalities. Previous works take the overall event labels to supervise both audio and visual model predictions. However, we argue that such overall labels harm the model training due to the *audio-visual asynchrony*. For example, commentators speak in a basketball video, but we cannot visually find the speakers. Thus, I tackle this issue by leveraging the cross-modal correspondence of audio and visual signals. We generate reliable event labels individually for each modality by swapping audio and visual tracks with other unrelated videos. If the original visual/audio data contain event clues, the event prediction from the newly assembled data would still be highly confident. In this way, the model would not be misled by ambiguous event labels. In addition, I propose the cross-modal audio-visual contrastive learning to induce temporal difference on attention models within videos, *i.e.*, urging the model to pick the current temporal segment from all context candidates. Experiments show the proposed method outperform state-of-the-art methods by a large margin.

## 1.2 Video Action Anticipation

Recent years have witnessed significant progress in the video analysis field. Advanced deep convolutional neural networks (CNN) models [98, 88] have achieved superior performance on the action recognition task, which is asked to classify the current action based on video content. However, there are few studies on predicting the future action seconds before it is executed. The action anticipation task, which requires the model to predict the future action that occurs several seconds later, is of vital importance in many real-world applications, such as house robots and autonomous driving.

In Chapter 5, I focus on the action anticipation task that predicts the future action based on past video content. Most existing works first summarize the current

video frames and then predict the next action based on current observations in a direct manner. Differently, I propose to first find some clues to generate the missing frames in the unobserved period. I thus decouple the action prediction task into several intermediate feature predictions, where I try to figure out the changes of frame representations in the future and then recognize the future action based on these intermediate pseudo representations. The intermediate feature prediction is conducted on multiple modalities of videos, *e.g.*, RGB feature, optical flow feature, and object detection features.

Different from other anticipation works that leverage intermediate representations, the model is trained by contrastive learning rather than regression. I design a new target to train the model by asking it to select the ground truth future features from all candidates. I also improve the model by introducing the residual connection in intermediate feature generating, where the model only needs to predict the difference of two frames rather than predict the whole representation. The residual design would encourage the model to pay more attention to the feature changes and transition from the past to the future. Experiments on large-scale video datasets show the proposed method significantly beats existing state-of-the-art methods.

## 1.3 Captioning for Unseen Objects

Language is also an important modality in our daily life. There are many vision and language multimodal tasks, such as image captioning, visual question answering, and visual dialog. Among them, video/image captioning aims at automatically describing images by sentences. It often requires lots of paired image-sentence data for training. However, theses trained captioning models can hardly be applied to new domains in which some unseen objects exist.

In Chapter 6, I introduce the zero-shot novel object captioning task, where the machine generates descriptions about unseen objects without extra training sen-

tences. To tackle the challenging task, I mimic the way that babies talk about something unknown, *i.e.*, using the word of a similar known object. Following this motivation, I utilize an external detection model to build a key-value object memory, containing visual information and corresponding words for objects in the image. For those unseen objects, I use words of most similar seen objects as proxy visual words to solve the out-of-vocabulary issue. I then propose a Switchable LSTM that incorporates knowledge from the object memory into sentence generation. The model has two switchable working modes, *i.e.*, 1) generating the sentences like a standard LSTM and 2) retrieving a proper noun from the key-value memory. The two modes are controlled by a switch indicator in the LSTM cell. Unlike existing works, the proposed model is learned to fully disentangle language generation from training objects, thus requiring zero training sentences in describing novel objects. Experiments on two large-scale datasets demonstrate the ability of the proposed method to describe novel concepts. Without extra training data, the proposed model even outperforms state-of-the-art methods (with additional training sentences) on the F1-score metric.

# Chapter 2

# Literature Survey

This chapter introduces a survey of related work in multi-modal perception and video analysis, including audio-visual video understanding, video action prediction based on multiple modalities, and vision-language applications.

I first introduce some advanced research for the video analysis in Section 2.1. Based on these video backbones, we start to address the audio-visual video understanding. Thus I introduce the literature review on audio-visual representation learning in Section 2.2. In Section 2.3, I review the typical methods working on the audio-visual events localization/parsing in videos. The advanced research for video action prediction is introduced in Section 2.4. In Chapter 4 and Chapter 6, the proposed Modality-aware model and the ImagineRNN are based on contrastive learning. Thus I introduce some related contrastive learning methods in Section 2.5. The related work on vision and language multi-modal perception is introduced in Section 2.6.

## 2.1 Video analysis with deep learning

Video analysis includes many tasks with video content, such as action recognition, event localization, and action anticipation. With the rapid progress of deep learning, there are many works achieving promising performance in understanding video content [88, 97, 8, 115]. The most general video analysis backbones are based on the action recognition models. In the early stage, researchers use 2D convolution neural networks for video analysis. Simonyan et al. [88] designed a two-stream

model, with one CNN model for RGB frames and the other for optical flow frames. Wang et al. [107] developed Temporal Segment Networks (TSN) for action recognition, which samples frames from different temporal segments and aggregates the prediction on each frame as the final prediction. Tran et al. [97] introduced 3D convolution for recognizing actions in videos. Carreira [8] proposed I3D backbone, which first initializes the 3D CNN model using the 2D CNN pre-trained weights from ImageNet. Our methods are based on these basic backbones for detailed multi-modal video analysis.

Different from the action recognition task, action localization [87, 56, 85, 128, 61, 127, 124] aims at localizing actions within untrimmed videos, where may contain additional information that is not relevant to the target classification. Previous supervised methods for action localization [85, 87, 128] usually first generate action proposal candidates and then predict the action based on these proposals. There are also weakly-supervised works [59, 69, 75, 90, 106] proposed for action localization. These methods usually use Multiple Instance Learning (MIL) for training without temporal boundary annotations. Nguyen et al. [69] proposed a sparse regularization to improve the action localization recognition. Shou [86] explored score contrast in the time axis.

## 2.2 Audio-visual Representation Learning

Recently, the multimodal research has been very popular and attracted lots of attentions [4, 5, 117, 72, 130, 73, 34, 131]. For these multi-modal research, some researchers focus on the sound and vision representation learning task. Most of these works assume that auditory and visual information are supposed to be synchronized in nature. Thus these two modalities can be leveraged for self-supervision in a cross-modal way. Based on this motivation, Owens *et al.* [73] proposed to take ambient audio as a strong supervision signal to guide the visual representation learning.

Arandjelovic and Zisserman [4] designed an auditory-visual correspondence proxy task to jointly train audio and vision features in the unsupervised learning way. Differently, Aytar *et al.* [5] developed SoundNet that uses a vision teacher model to learn audio feature using unlabeled videos.

Some other research leverages the temporal synchronization of audio and visual signals for representation learning. Korbar et al. [50] designed to learn auditory and visual features by the self-supervised vision-sound temporal synchronization task. In the proxy task, the model is trained to predict if the visual frames are temporally synchronized with the audio signals.

Besides audio-visual representations learning, there are many audio-visual applications such as sound separation [16, 23, 27, 28, 29, 125, 126, 129], sound source localization [71, 83, 95], audio-visual action recognition [30, 46], audio-visual navigation [9, 24], audio-visual video captioning [77, 92, 93, 108], and audio-visual event localization [58, 95, 96, 119].

## 2.3 Audio-visual Events in Videos

Audio-visual Event localization aims at detecting and localizing audible or visible events from untrimmed videos. We introduce two kinds of audio-visual event localization in videos, including synchronized audio-visual events and asynchronous audio-visual video parsing.

Recently, Tian et al. [95] firstly propose the audio-visual event localization task, where the event includes audible objects. In [95], Tian et al.designed the sound-guided vision attention, which emphasis vision regions by the auditory features. Based on [95], Lin et al. [58] further designed to use a sequence-to-sequence model to aggregate auditory and visual representation into a global feature. These existing works only aggregate representations from the two modalities at the clip level.

The Audio-visual video parsing [94, 116] (AVVP) task is designed to provide the temporal localization analysis of audio and vision events from untrimmed videos. It would analyze videos by dividing them into a series of events with their event class, starting and ending points, and audio or visual modality information. Different from these works [58, 92, 119] targeting audio-visual event localization, in the AVVP task, the model is asked to predict all event information in both audio and visual track.

## 2.4 Video Action Prediction

Recently, many researchers [53, 55, 110, 122] study the video action prediction task, which is to predict the near future action given current observation. Existing works usually take a recurrent neural network to encode existing past frames [2, 26]. Miech et al. [66] developed an anticipation framework that directly anticipates future action based on the combination of past visual inputs and past action recognition results with the help of the action transition model. RULSTM [20] contains two Long short-term memory (LSTM) models to predict future actions, with one for summarizing the past, and the other for predicting the future action.

## 2.5 Contrastive Learning

Contrastive learning aims at optimizing models by distinguishing similar and dissimilar data pairs. Recent works [11, 33, 70] proposed to utilize contrastive learning for self-supervised learning. Contrastive Predictive Coding (CPC) [70] proposed to learn representation by encoding predictions over future observations from the past. MoCo [35] designed a momentum encoder and maintained a queue of representations to conduct contrastive learning. SimCLR [11] experiments with different combinations of data augmentation methods for paired samples in contrastive learning. Very recently, Han et al. [33] proposed to introduce contrastive learning into the action recognition task. The model is optimized by a predictive attention mechanism over

the compressed memories that predicts future representations based on recent observation. Different from these methods, we focus on the action anticipation task rather than representation learning. We found the contrastive learning helps to learn the change of future features, which can be used to obtain better intermediate imaginary data in our ImagineRNN framework.

## 2.6 Vision and Language

As an important modality in our daily life, natural language convey semantic information of our human description. Therefore, in the multi-modal learning field, vision and language have been popular in recent years.

Among these vision and language studies, one of the popular applications is automatically generating natural language sentences from images/videos, *i.e.*, image/video captioning. The task is to describe an image/video in natural sentences. Most of existing works are built on the encoder-decoder framework for learning the joint distribution of vision and text features [7, 15, 130, 45, 48, 65, 78, 103, 111, 112, 121, 113, 132]. In the encoder-decoder framework, the encoder is usually a visual CNN for encoding the input vision data into the feature space. The decoder is usually a recurrent neural network (RNN) that recurrently generates the next word by taking the last word as input. Vinyals et al. [103] use a vision CNN as an encoder and an RNN as a decoder to generate language. The whole framework is trained in an end-to-end manner. The framework was improved using the attention module [121], which encourages the model to pay attention to the important image regions in predicting the words. However, the methods could only generate sentences for seen objects, since the vocabulary of the model is fixed during training. There is little extension capacity of these models.

In this thesis, I focus on the novel object captioning task, which is designed to generate sentences for both seen and unseen objects. Henzdricks et al. [37] use

an ImageNet pre-trained classification model and a language model pre-trained on additional language datasets. Based on [37], Venugopalan et al. [100] proposed to improve the model by joint training the vision, language, and captioning modules. Yao et al. [123] proposed LSTM-C that copies the object detection prediction in generating the words by the language model.

# Chapter 3

# Double Attention Corresponding for Audio-Visual Event Localization

In this chapter, I investigate the synchronous audio-visual event localization task. The core idea of our proposed DAC is that we first look at the whole video to obtain the global knowledge, and then attain local event clues by the global-to-local cross-modal comparison. The motivation is that we believe we humans should see the global video to access the overall event information, and look into each clip in detail with the guidance of global concept.

## 3.1 Introduction

We study the audio-visual event localization task, where we regard an event that is both audible and visible as the audio-visual event. Our target is to find the boundary along with the temporal axis and recognize the class of the ongoing event. It consists of two subtasks, *i.e.*, the *cross-modality localization* (CML) task and the *supervised audio-visual event localization (SEL)* task.

As shown in Fig. 3.1, in the CML task, the model needs to localize the vision frames in the temporal axis from input audio and visual versa. CML is useful when the user wants to find an event in aerial videos by inputting audio, since aerial videos usually do not have sound. The task requires the model to be generic without accessing the event classes. Therefore, the task focuses on the generalization ability to unknown testing. In the SEL task, the model should find which temporal clip of the video contains the audio-visual event and its event class.

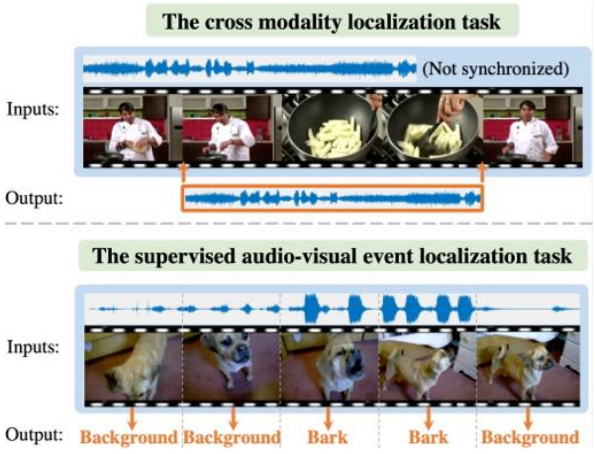


Figure 3.1 : The visual explanation of the audio-visual event localization task.

Existing works [95, 58] split the video into a series of small clips, and then extract local CNN features for the audio and vision frames in each clip. These methods optimize the features of two modalities to be close or concatenate audio and visual features in the clip level.

Although the clip-level operation is useful for temporal localizing events, the local clip is too short and contains a partial observation of the event. The sound and vision signals may change a lot during the event period. Besides, concatenating vision and sound representations at the clip level would be very fragile to incremental misplacements or noise. In conclusion, existing works leverage the local auditory-visual correlations, but miss the global one.

For an event video, there are clues in sound and vision modalities about the ongoing event, *e.g.*, seeing a little baby and hearing crying concurrently. The joint co-occurrence of audio and vision implies that it is an event, since it is improbable that they occur in both modalities merely by chance. This motivates us to leverage the global co-relation between audio and vision signals in event localization.

Therefore, we design the Double Attention Corresponding (DAC) method, which

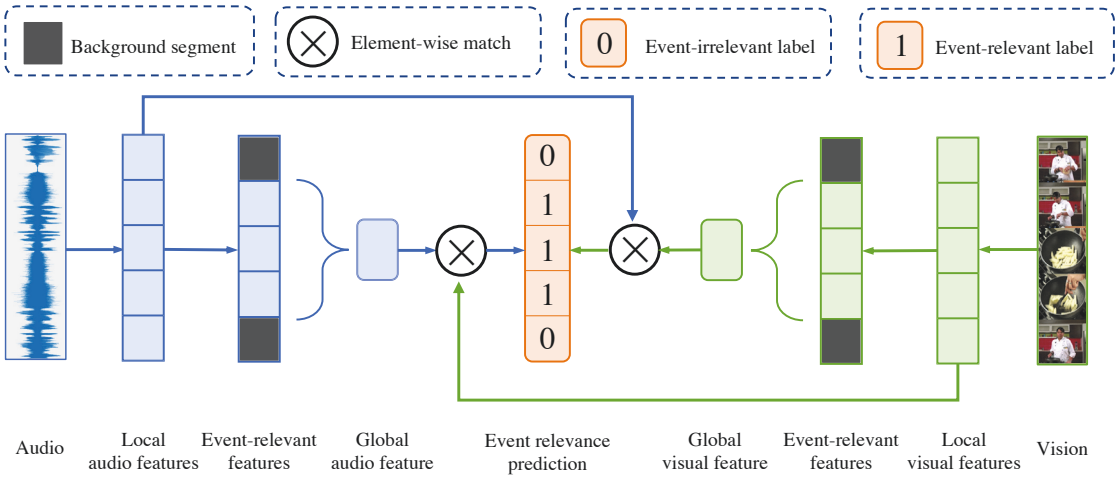


Figure 3.2 : The proposed dual attention matching (DAC) module.

looks at the whole video to obtain the global knowledge, and then attains local event clues by the global-to-local cross-modal comparison. The motivation is that we believe we humans should see the global video to access the overall event information, and look into each clip in detail with the guidance of global concept. We take the overall feature of one modality to query the clip feature in the other modality, and find which clip in the target modality is close to the overall event.

Our DAC can be used in both CML and SEL tasks. Experiments validate that our DAC significantly beats the state-of-the-art methods.

## 3.2 Methodology

### 3.2.1 Preliminaries

Denote the $N$-seconds video to be $X = (X^A, X^V)$, where $x^A$ indicates the sound, while $x^V$ is the vision track. We divide the video into $N$ clips $\{x_t^A, x_t^V\}_{t=1}^N$ with each clip lasting one second.

$x_t^A$ and $x_t^V$ is the sound and vision clip at $t$-th second, respectively. For a syn-

chronized sound-vision combination $(x_t^A, x_t^V)$, its event relevant label $y_t \in \{0, 1\}$ means the relevance of the audio and visual modalities about the event, where we use $y_t = 1$ to indicate the sound $x_t^A$ and vision clip $x_t^V$ is related to the event. Also, we have the *event-relevant region* $T_E = \{t|y_t = 1, 1 \leq t \leq N\}$, indicating the temporal boundary of the event. We use pre-trained CNN models to extract local features for each clip, denoted as $f_t^A$ and $f_t^V$ for sound and vision, respectively.

### 3.2.2 Double Attention Corresponding

We leverage a global aggregation on the event clips $T_E$ for obtaining the overall event representations. Then the inner product of global and local features is used as the cross-modal correspondence. We use the event relevance label $y$ to optimize the output. In this way, the local features of the event-related region are trained to be close to the global event feature from the other modality. And for the background clips where no event happens, the model would push these local features far away from the global event representation. We show the framework in Fig. 3.2.

Our DAC contains two components, *i.e.*, global event modeling, and the cross-modal corresponding.

**Global event modeling.** We use self-attention as the global event modeling on the event-related clips. The attention mechanism is,

$$\texttt{att}(q, k, v) = \texttt{Softmax}(\frac{qk^T}{\sqrt{d}}, v), \tag{3.1}$$

where $q$, $k$, $v$ indicates the query, key, and values with dimension $d$. They are created by transformations of the input feature,

$$\texttt{self-att}(x) = \texttt{att}(W_q x, W_k x, W_v x). \tag{3.2}$$

$W_q$, $W_k$, and $W_v$ are the linear weights for query, key, and values. Then we average pool the self-attention output as the final event representation in the modality.

For the segments in the event region $T_E$, we have the concatenated tensor of local features, $F_E^A = \{f_t^A | t \in T_E\}$ and $F_E^V = \{f_t^V | t \in T_E\}$. Thus we can obtain the event representation via,

$$\phi^A(X^A) = \texttt{avg}(\texttt{self-att}(F_E^A)), \phi^V(X^V) = \texttt{mean}(\texttt{self-att}(F_E^V)). \tag{3.3}$$

where `avg` indicates averaged pool in the temporal axis.

**Cross-Modal Double Corresponding.** We believe information should be different in the event clips and background clips. Thus we train our model to pick which clip is related to the event in the other modality. The relation/similarity is defined by the dot product of the event feature (from one modality) and the clip-level features (from the other modality). Thus the mechanism is applied on both modality sides (cross-modal check). The event-related prediction $p_t$ can be obtained by,

$$p_t^A = \texttt{Sigmoid}(\phi^V(X^V) \cdot f_t^A), \tag{3.4}$$

$$p_t^V = \texttt{Sigmoid}(\phi^A(X^A) \cdot f_t^V), \tag{3.5}$$

$$p_t = \frac{1}{2}(p_t^A + p_t^V). \tag{3.6}$$

$p_t^A$ and $p_t^V$ is the event-related prediction on the audio modality and the vision modality, represetively. Since this is a binary classification problem, we use the Sigmoid function instead of cosine distance calculation for the binary prediction here.

We use the event-relevant label $y_t$ to optimize the event-relevant prediction. We expect the model prediction $p_t$ should be close to 1 for the event-relevant clip ($t$ is in the event-relevant region $T_E$), and 0 for the background region. The training is optimized using the Binary Cross Entropy (BCE) loss.

### 3.2.3 Cross-Modality Localization

Next, we illustrate how we apply our DAC in the CML task.

The target is to localize the position of the synchronized clip in one modality using the input of the event-based clips from the other modality. Taking the audio to vision (A2V) direction as an example, the input is a $b$-second event-related audio $\hat{X}^A$. The target is to localize the counterpart $b$-second vision clip from the whole vision frame sequences $\{s_t^V\}_{t=1}^N$.

Our trained DAC model can be directly used in the inference of the CML task. We use the input query global feature to check each local clip in the other modality, thus we can have the prediction score for each clip, which indicates the related prediction about the query. After that, we use a sliding window to calculate the prediction score sum of all the $b$-second candidates from the whole $N$-second sequences. The final localization prediction is the $l$-length segment with the largest score sums.

During training, the videos of the training datasets are normal (aligned video/audio). We split the video into a series of local segments (one second long). We first extract the global event feature of one modality, and then calculate the event relevance predictions on each local segment in the other modality. If the local segment also belong to the event region, the ground truth label is 1 here; otherwise it is 0. In this way, even though we do not create misplaced audio during training, we can leverage all the audio segments and ask the model to classify which segments is related and which one is unrelated.

### 3.2.4 Event Localization in Audio-Visual Videos

In the audio-visual event localization task, we leverage the event-relation annotations $y$ and its class label $y^c$. We assume that there is only one event existing in each video. We need to recognize the class (including background) for all clips of the whole video.

Different from existing works [95, 58], we split the framework into two stages. We first predict the event class based on the event representation, and then find

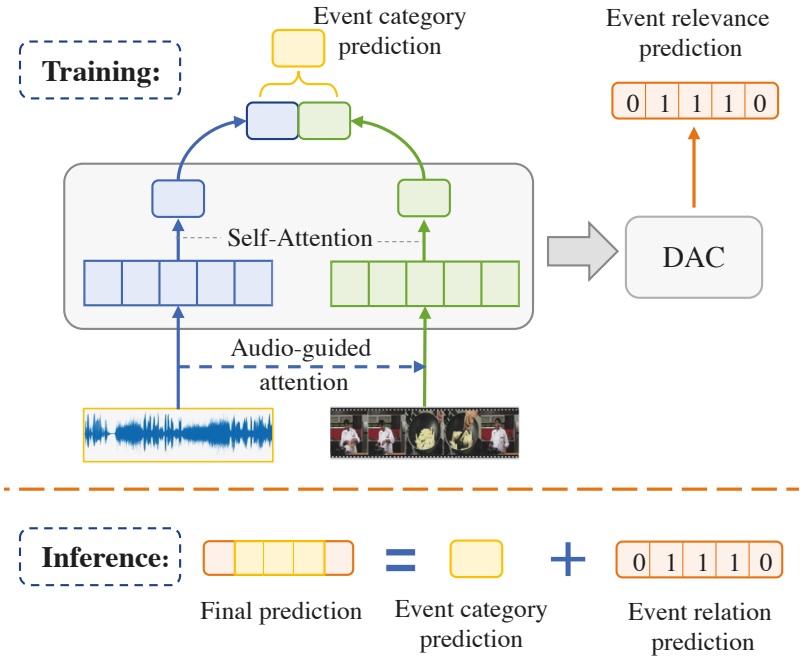


Figure 3.3 : The pipeline for the Event Localization task.

foreground/background clips. We show our model in Fig. 3.3. Since we do not have the event region boundary annotation $T_E$ in testing, we input all clips including those background clips to the self-attention model. We then concatenate the two event representations of audio and vision modalities to predict the event class $\hat{y}^c$.

At the same time, our DAC module would use the event representations to check each clip-level local region to predict the event relevance $\hat{y}_t$. We use the prediction to decide whether to change the prediction on the $t$-th clip to the background class. For the $t$-th clip, the final prediction is background if $\hat{y}_t < 0.5$. Otherwise, we use the event class prediction $\hat{y}^c$ as the final prediction.

The training loss function is,

$$\mathcal{L} = \lambda \mathcal{L}^c + (1-\lambda)\frac{1}{N}\sum_{t=1}^{N}\mathcal{L}_t^r, \tag{3.7}$$

where $\mathcal{L}_t^r$ denotes the Binary Cross Entropy loss for the event relation output $\hat{y}_t^r$. $\mathcal{L}^c$ is the Cross-Entropy loss for the event class output $\hat{y}^c$. $\lambda$ is the weight that balances

| Models | A2V Accuracy | V2A Accuracy | Avg Accuracy |
|---|---|---|---|
| DCCA [3] | 0.341 | 0.348 | 0.345 |
| AVDLN [95] | 0.356 | 0.448 | 0.402 |
| **Ours** | **0.471 ± 0.016** | **0.485 ± 0.014** | **0.478 ± 0.015** |

Table 3.1 : Results on the CML task.

the two losses.

## 3.3 Experiments

### 3.3.1 Experiment Settings

**The Audio-Visual Event (AVE) dataset** [95] is a subset from AudioSet [31]. The AVE dataset has 4143 audio-visual videos, including 28 classes of events, which covers a lot of human activities. The length is ten seconds for each video.

**Evaluation metrics.** The CML task has two directions, *i.e.*, using audio to localize video clip (A2V) and using video to find audio clip (V2A). We define a correct matching is that the localized clip is exactly the ground truth. Other predictions are regarded as the missing match. We use the rate of correct matching overall evaluation data as the accuracy in evaluating the CML model. As for the SEL task, the averaged recognition accuracy of all clip-level predictions is used to evaluate the model.

**Implementation details.** We take ImageNet-pretrained VGG-19 [89] model to extract visual features for each clip. For audio, we use the AudioSet [31] pretrained VGG-like model [38] to obtain audio representation. For a fair comparison, we leverage the same local clip-level features as DCCA [95]. For simplicity, we don't add any position encoding embedding modules in our DAC model.

| Method | Accuracy |
|---|---|
| ED-TCN [51] | 0.469 |
| AVE [95] | 0.714 |
| AVSDN [58] | 0.726 |
| AVE+Att [95] | 0.727 |
| Ours DAC | 0.745 |

Table 3.2 : Results of the SEL task on the AVE dataset.

### 3.3.2 Comparison with Existing Works

In Table 3.1, we show the comparison of our model and state-of-the-art works on the CML task. The compared method AVDLN only operates at the local clip-level, and models the relation between audio and visual modalities by the Euclidean distance. Differently, we first watch the whole event video to obtain a better event feature and then look into each clip for detailed localization. Thus our model outperforms these competitors by a large margin, as shown in Table 3.1.

On the SEL task, we compare the DAC model with the following state-of-the-art works. ED-TCN [51] is the state-of-the-art method for temporal action localization. Tian et al. [95] use the LSTM to build temporal modeling for joint modalities. Then they use the audio-guided visual attention module to automatically learn which visual part is useful for the audible object. AVSDN [58] introduces an extra LSTM for replacing the previous event classifier. We compare our DAC to these models in Table 3.2, and observe that our DAC achieves higher accuracy compared to them.

### 3.3.3 Ablation Studies

**Different temporal modeling methods.** We conduct ablation studies to study different temporal modeling methods (replacing the self-attention model) used in

| Method | V2A Accuracy | A2V Accuracy | Avg Accuracy |
|---|---|---|---|
| DAC w/ RNN | 0.418 | 0.479 | 0.449 |
| DAC w/ Averaged Pooling | 0.460 | 0.461 | 0.461 |
| DAC w/ Max Pooling | 0.458 | 0.462 | 0.460 |
| DAC w/ LSTM [39] | 0.435 | 0.481 | 0.458 |
| DAC w/ GRU [12] | 0.455 | 0.474 | 0.465 |
| DAC w/ BLSTM [82] | 0.442 | 0.481 | 0.462 |
| DAC w/ Self-Att [99] | 0.471 | 0.485 | 0.478 |

Table 3.3 : Ablation studies of temporal modeling modules used in DAC on the CML task.

our DAC. We test several common temporal functions, such as Averaged Pooling, Max Pooling, RNN, LSTM, Bidirectional-LSTM, GRU, and Self-Attention. The results are shown in Table 3.3. We found self-attention leads to the highest score compared to other modeling functions. Even with naive modeling methods (e.g., Averaged Pooling), our DAC still beat the state-of-the-art clip-level model [95]. We cannot learn such a cls token as usually done in transformer structures, since we need a temporal modeling module to gather the global event information.

The reason for such a significant improvement is the usage of global event information. If we take the overall event features as query, the model could be more clear about the ongoing events, leading to a better localization performance in the downstream tasks. The core reason behind also proves our motivation that the model should perceive the whole event video before looking into each local clip.

**Cross-modal checking versus self-checking.** We investigated the cross-checking mechanism of DAC. We tried to replace the cross-modal event feature in Eqn. 3.4 and Eqn. 3.5 with the event feature of the same modality. For example, instead of global

| Models | V2A Accuracy | A2V Accuracy | Avg Accuracy |
|---|---|---|---|
| Ours (Self-checking) | 0.286 | 0.298 | 0.292 |
| Ours (Cross-checking) | 0.471 | 0.485 | 0.478 |

Table 3.4 : Ablation studies on self-checking and cross checking on the CML task.

| Models | Acc |
|---|---|
| DAC (No checking) | 0.707 |
| DAC (Self-checking) | 0.742 |
| DAC (Cross-checking) | 0.745 |

Table 3.5 : Ablation studies on the matching mechanism on the SEL task.

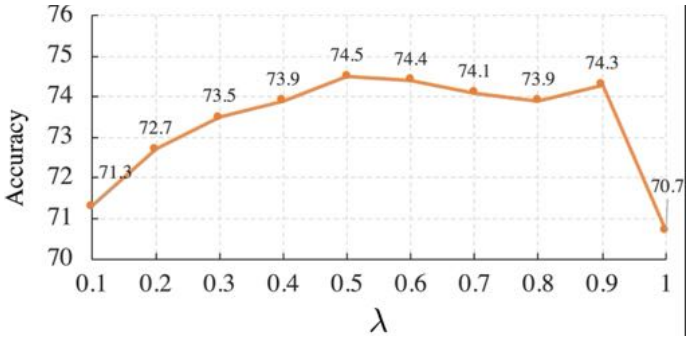


Figure 3.4 : Analysis on different balancing weights $\lambda$.

event features from the visual channel, we use audio global event representation to check audio clips. Table 3.4 shows the results on CML. "Ours (Self-checking) ' is the model mining weak correspondence from the self modality. We can see that the results are much weaker than our DAC. It validates us that the cross-modal temporal relation is a strong signal for CML.

We also report the comparison of self-checking and cross-checking on SEL. The

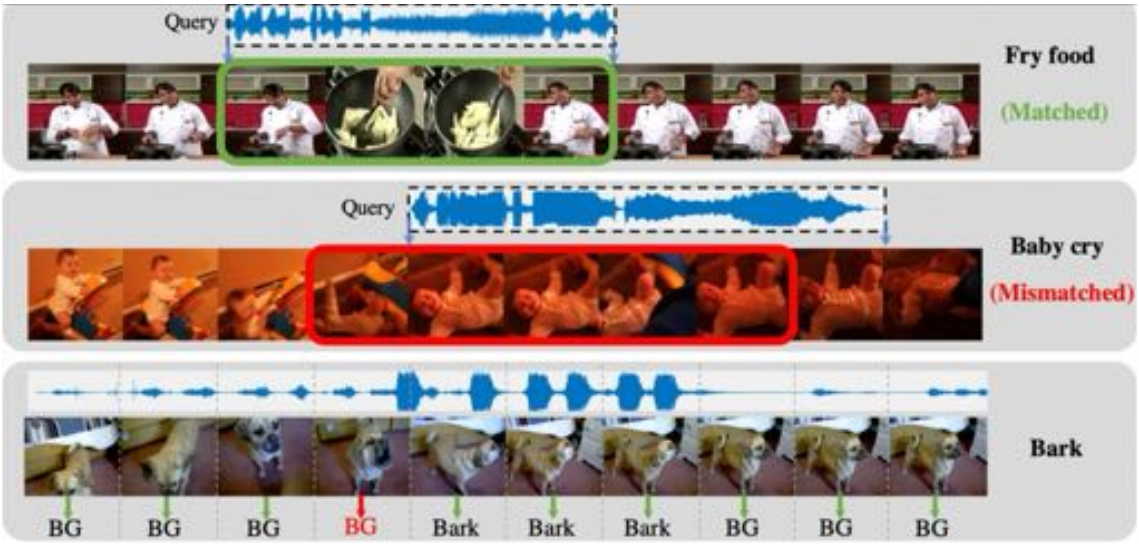


Figure 3.5 : Qualitative results. We use green to indicate the correct prediction and red for error predictions. The upper two blocks is the CML task, and the last one is the sample of SEL task.

model "DAC (No checking)" is the one that does not use the overall event feature for checking each local clip. The performance of the model is poor (0.707), since it could not distinguish the event region and background clips. The model with self-checking improves the above one by 0.035 in accuracy. Our final DAC with cross-checking leads to the best performance by using cross-modal interaction.

**Impact of the balancing weight** $\lambda$. In our model, $\lambda$ is a weight that balances the ratio of contribution brought by the relation loss $\mathcal{L}^r$ and the classification loss $\mathcal{L}^c$. We study different values of the combined weight, and show the results in Fig. 3.4. If we set $\lambda$ to be 1, it means only classification loss is used for training. We observe the best accuracy is obtained with $\lambda = 0.5$.

### 3.3.4 Visualization Examples

Some qualitative examples of DAC have been shown in Fig. 3.5. We use the green color to indicate the correct prediction, and the red color for error estimation.

The above two samples is for the CML task (audio to vision). The ground-truth position of the auditory query is shown in the temporal axis. The second row is a hard example. The input audio is a baby crying sound. Our model failed in localizing the vision clips (the red box). The last row is the results on SEL. Our DAC failed in the fourth clip, where DAC prediction is the background class, but the label should be "Bark" at this temporal clip.

## 3.4 Summary

In this chapter, we study the audio-visual video understanding problem. We focus on the event localization task, where auditory and visual event exists. Unlike existing works using local fusion only, we propose exploring the whole event video for better event modeling. After that, we cross-check all local clips to attain temporal localization information. Our designed DAC model takes the overall event feature from a modality to query all the clips in the other modality. We expect those event regions should have higher similarity (correspondence) scores. In this way, our model could obtain better audio-visual cross-modal ability in video understanding. Experiments also validate the effectiveness of our DAC model.

# Chapter 4

# Exploring Heterogeneous Clues for Weakly-Supervised Audio-Visual Video Parsing

In the previous chapter, I introduce the proposed Double Attention Corresponding model for audio-visual event localization, which takes global event information to check each local clip in a cross-modal way. However, the model is designed for synchronized audio-visual events. In this chapter, I study a more general audio-visual video understanding task, where the audio event and visual events may not align well.

## 4.1 Introduction

We humans explore and perceive the sounding environments with sensory streams, including visual, auditory, tactile, etc.. Among these simultaneous sensory streams, visual information and auditory information are two fundamental streams that widely contain massive information in our daily life.

Audio-visual comprehension [58, 95, 25, 94, 119] is more robust in identifying the ongoing events compared to those vision models [107, 87]. For example, occlusions and blind spots are common in egocentric videos and web videos, where the target object is outside of the view. In such situations, auditory signals could provide reliable clues for video understanding.

Existing audio-visual research works [4, 16, 23, 27, 29, 40, 22, 50, 71, 83, 125, 126, 129] usually assume vision and sound data are always temporally matched. However, this alignment might not always hold in practice. We may find lots of

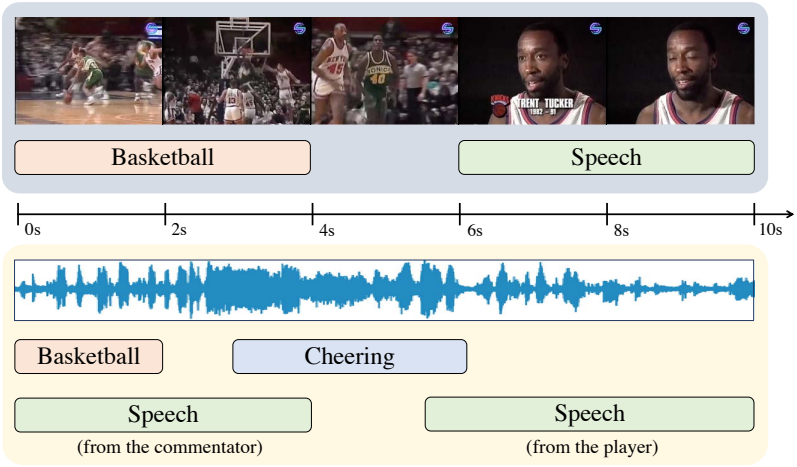


Figure 4.1 : Examples of the audio-visual video parsing task. Colored rectangles indicate the ground truth events. Taking the visual and audio data as input, we aim at identifying the audible and visible events and their temporal location. Note that the visual and audio events might be asynchronous.

videos whose sound originates outside of the scene view. Despite the nonalignment, audio signals are still important in understanding the events, such as out-of-screen motorcycle racing. In this chapter, we study the audio-visual video parsing (AVVP) task [94], which aims at providing a detailed analysis of auditory, visual, and audio-visual events in videos without such alignment assumptions. As shown in Fig. 4.1, the target of AVVP is to recognize event categories in each sensory modality and localize them temporally in videos.

Due to exhausting labeling cost, Tian et al. [94] proposed a weakly supervised learning framework, which does not require dense annotation but only use sparse labeling for training (the presence or absence of event categories). The weakly supervised labels only indicate which event occurs in the video, without detailed modalities and temporal boundaries. The weakly supervised labels are more comfortable to obtain and could be boosted with automatic annotation (tags) for web videos. To

solve the challenging issue, Tian et al. [94] proposed introducing cross-modal and self-modal attention to obtain aggregated features. The model is optimized in the Multimodal Multiple Instance Learning (MMIL) way, which regards overall event labels as the optimization targets for both audio and visual predictions.

However, audio and visual content are naturally different sensory streams. Visual data are captured by specific camera views, while audio signals collected by microphones could perceive all audible events of the scenes. Unlike other weakly supervised learning tasks, some event information may only exist in a single modality (either audio signals or visual signals). It would be irrational to optimize both modality predictions to be close to the overall event labels. For example, in a basketball match video, there might be commentators speaking, but we cannot find them visually (see Fig. 4.1). It harms the visual model optimization if we follow the universal weakly supervised learning way.

In this chapter, we tackle the challenging AVVP task by exploring heterogeneous clues. We alleviate the modality uncertainty issue and generate reliable event labels individually for each modality without additional annotations. To achieve the goal, we exchange the audio and visual track of a training video with other unrelated videos. Our motivation is that the newly assembled video's prediction would still be highly confident if the visual/audio signals do contain clues of the target event. Otherwise, the event information is not visible/audible in the corresponding modality. In this way, we could obtain precise modality-aware event labels and protect models from being misled by the ambiguous overall labels. To the best knowledge of ours, we are the first that swap audio and visual tracks with other videos to assess the modality uncertainty.

In addition, we also propose to induce temporal difference within videos in a contrastive learning manner. Previous methods obtain enhanced modality features

by leveraging all temporal contexts of the whole video. We argue that these might harm the model performance since it obscures the temporal difference within an event video. Since we do not have temporal annotations in training, inspired by self-supervised learning [35, 120], we propose to introduce contrastive learning to introduce temporal difference into aggregated features. We urge the attention model to pick the correct temporal cross-modal segment features from all candidate distractors. Thus the aggregated feature would be closer to the current segment instead of all context segment, leading to better temporal localization performances.

The contributions are listed as follows. We propose to address the modality uncertainty issue by exchanging audio and visual tracks with other videos. Thus we can obtain accurate modality-aware event supervision instead of ambiguous overall labels. We further introduce temporal heterogeneous constrain into the attention model via contrastive learning, which alleviates the ambiguous temporal boundaries issues in the weakly-supervised AVVP task. Experiments prove our model significantly beats the state-of-the-art works on all evaluation metrics. Specifically, we improve the segment-level audio-visual parsing accuracy from 48.9% to 55.1% on the LLP dataset.

## 4.2 Method

### 4.2.1 Preliminaries

**Problem statement.** In the AVVP task, each video may contain multiple visible or audible events. Note that many events may only exist in one modality (either audio signals or visual signals). For a $T$-seconds audio-visual video sequence $\mathcal{S} = \{V_t, A_t\}_{t=1}^{T}$, $A$ is the audio track and $V$ is the visual counterpart at the $t$-th segment. Each segment lasts for one second long. For *evaluation*, the targets are to predict the event labels for each segment and each modality. For the $t$-th video segment $(V_t, A_t)$, the target $\mathbf{y}_t = (y_t^a, y_t^v, y_t^{av})$ is a multi-class event label. Note there may

exist zero or many events that are happening at the $t$-th moment. $y_t^a$ indicates the audio event label. $y_t^v$ is the event label in the visual channel. $y_t^{av}$ is audio-visual event labels, which means events are both audible and visible simultaneously.

For *training*, we only have access to weakly-supervised labels. Specifically, we only know events that show up in the video sequence $\mathcal{S}$, but *do not* have precise labels such as the events occurring time and modalities. Therefore, the temporal and multimodal uncertainty in the weakly-supervised AVVP problem makes it very challenging.

**Data process.** Pre-trained audio and visual deep models are applied to obtain visual representations $\{\mathbf{f}_t^v\}_{t=1}^T$ and audio representations $\{\mathbf{f}_t^a\}_{t=1}^T$ at the segment level (one second per segment), respectively. The extracted audio and visual features are used as input for the following modeling.

**Feature aggregation.** Previous work [94] proves the effectiveness of feature aggregation upon the local input features. Thus we also enhance the input features by leveraging context information via self-attention and cross-attention mechanism. $\texttt{att}(\cdot)$ is defined by,

$$\texttt{att}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \texttt{Softmax}(\frac{\mathbf{q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{4.1}$$

where $d$ means the dimension of the feature vector $\mathbf{q}$. The aggregated feature can be obtained by,

$$\hat{\mathbf{f}}_t^a = \mathbf{f}_t^a + \texttt{att}(\mathbf{f}_t^a, \mathbf{F}^a, \mathbf{F}^a) + \texttt{att}(\mathbf{f}_t^a, \mathbf{F}^v, \mathbf{F}^v), \tag{4.2}$$

$$\hat{\mathbf{f}}_t^v = \mathbf{f}_t^v + \texttt{att}(\mathbf{f}_t^v, \mathbf{F}^v, \mathbf{F}^v) + \texttt{att}(\mathbf{f}_t^v, \mathbf{F}^a, \mathbf{F}^a), \tag{4.3}$$

where $\mathbf{F}^a = (\mathbf{f}_1^a, ..., \mathbf{f}_T^a)$ and $\mathbf{F}^v = (\mathbf{f}_1^v, ..., \mathbf{f}_T^v)$ are the auditory and visual features sequence from the video $\mathcal{S}$, respectively. For simplicity, we assume that the dimensionality of audio and visual features are the same during the feature aggregation.

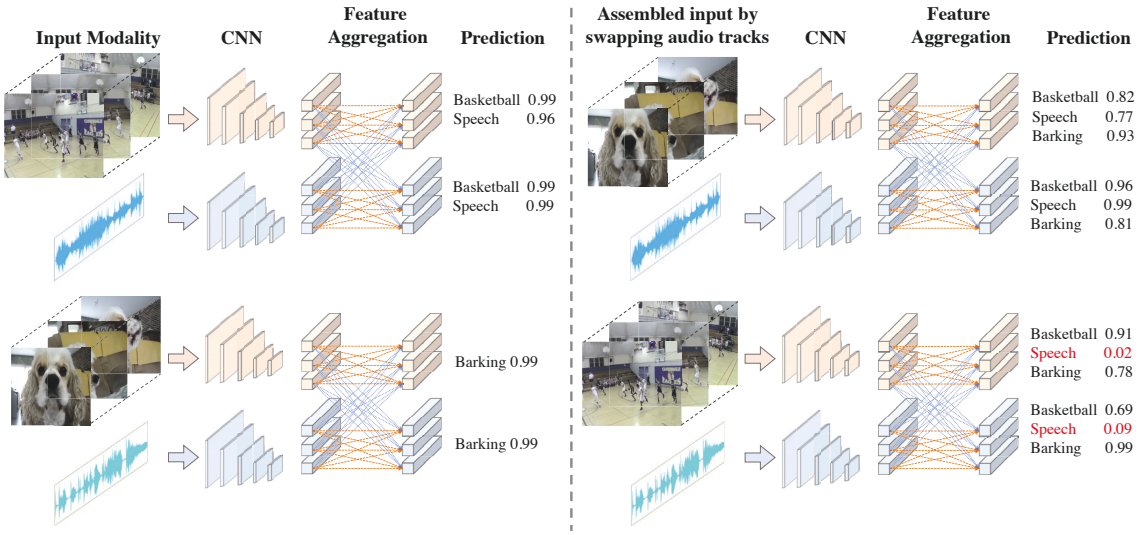


Figure 4.2 : The modality-aware label refining (MA) pipeline.

Compared with the original input features, the aggregated features $\hat{\mathbf{f}}_t^a$ and $\hat{\mathbf{f}}_t^v$ are promoted by gathering event information across the entire video content.

**Multiple Instance Learning.** The event prediction of each segment and modality is based on the aggregated features. Since there might be multiple events happening at the same segment, we use the Sigmoid activation on the classifier for outputting probability for each event category. We denote $p_t^a$ and $p_t^v$ to be the audio event predictions and the visual event predictions at the $t$-th segment, respectively. However, we only have the global weakly supervised label $\bar{\mathbf{y}}$ instead of accurate segment-level labels in the weakly-supervised training. Following [94], we take the attentive MIL pooling in predicting the video-level events. The overall event predictions $\bar{p}^a$ and $\bar{p}^v$ are obtained by the weighted average of all segment-level predictions. Specifically, to compute the attention weights, we use a fully-connected layer to transform all the frame-level prediction. Then the weights to snippets at different time steps is calculated by the temporal attention mechanism, which is a softmax operation over all the transformed temporal tensors. For our baseline, we optimize the video-

level probability $\bar{p}^a$ and $\bar{p}^v$ to be close to the overall event labels $\bar{y}$ using the binary cross-entropy loss function.

### 4.2.2 Exchanging Audio and Visual Tracks

The above baseline could be used to train a decent model for weakly-supervised AVVP. However, it may induce severe label noise due to the modality uncertainty. Many events may only exist in one modality (either audio signals or visual signals) since audio and visual content are naturally different information sources. Optimizing both modality predictions (*i.e.*, $\bar{p}^a$ and $\bar{p}^v$) to be close to the overall labels would inevitably introduce noise in training.

Motivated by the natural correlation between audio and visual content, we propose alleviating the modality uncertainty issue by exchanging audio and visual tracks with other videos. As shown in Fig. 4.2, we first assess modality uncertainty and then generate modality-aware event labels for each modality individually. Finally, we re-train our model from scratch based on these refined labels.

**Exchanging channels.** Our target is to localize the target event between modalities, *i.e.*, whether a modality contains the target events or not. To achieve the goal, we leverage other videos to assess the target video without requiring additional annotations. Suppose we have two audio-visual videos that have disjoint video-level event labels, *i.e.*, $\mathcal{S}^i = (V^i, A^i)$ and $\mathcal{S}^j = (V^j, A^j)$, but $\bar{\mathbf{y}}^i \neq \bar{\mathbf{y}}^j$. Taking the video $\mathcal{S}^i = (V^i, A^i)$ as our target video, we exchange the visual channel and audio tracks of these two videos and form a new "video" by,

$$\hat{\mathcal{S}}^i_j = (V^i, A^j), \tag{4.4}$$

$$\hat{\mathcal{S}}^j_i = (V^j, A^i), \tag{4.5}$$

where $\hat{S}^i_j$ denotes the new "video" formed by the visual content from the video $\mathcal{S}^i$ and the audio track from the video $\mathcal{S}^j$. Since the video-level event labels $\bar{\mathbf{y}}^j$

guarantee there is no event $\bar{\mathbf{y}}^i$ existing in any modality of video $S^j$, we could safely conclude that both $V^j$ and $A^j$ are unrelated to the target event $\bar{\mathbf{y}}^i$. Thus for the newly assembled data $\hat{\mathcal{S}}_j^i$ and $\hat{\mathcal{S}}_i^j$, the only clues about the event information $\mathbf{y}^i$ are from the content of $i$-th video $\mathcal{S}^i$, *i.e.*, either from $V^i$, $A^i$ or both.

**Assessing modality uncertainty.** We assume that the newly assembled video's prediction would still be highly confident if the visual/audio signals do contain clues of the target event. In other words, the event information is likely to be *missed* in the remaining modality if the prediction is low on the assembled videos. Denote the base model to be $\phi(\cdot)$, we obtain the event predictions for these assembled videos by,

$$p_{\hat{a}}^v, p_{\hat{a}}^a = \phi(V^i, A^j)/E_c, \tag{4.6}$$

$$p_{\hat{v}}^v, p_{\hat{v}}^a = \phi(V^j, A^i)/E_c, \tag{4.7}$$

where $p_{\hat{a}}^v$ indicates the event prediction based on aggregated visual features for the video with *changed audio*, and $p_{\hat{v}}^v$ means the event prediction based on aggregated visual features for the video with *changed vision*. $E_c$ is the normalized error rate of the target event category $c$ according to training predictions. The intuition is that the misaligned labels are more likely to happen if we found it hard to optimize the corresponding event categories (training accuracy on event category $c$ is lower). We believe the predictions $p_{\hat{a}}^v$ and $p_{\hat{a}}^a$ indicate the reliability of event labels for the *visual* track in video $S^i$. Similarly, $p_{\hat{v}}^v$ and $p_{\hat{v}}^a$ are used to validate the reliability of event labels for the *audio* track.

**Refining modality-aware event labels.** By assessing each modality's confidence, we could further refine the event labels and have different event labels for the two modalities. We reassign the event label and remove unrelated labels for each modality if the confidences are lower than a threshold 0.5, since the sigmoid prediction ranges from 0 to 1. Specifically, we would discard the event labels for *visual* modal-

ity if $p^v_{\hat{a}} < 0.5$ and $\hat{p}^a_{\hat{a}} < 0.5$. Similarly, we would also remove the event labels for *audio* modality if $\hat{p}^v_{\hat{v}} < 0.5$ and $\hat{p}^a_{\hat{v}} < 0.5$. We could roughly estimate whether the event happens visually or audibly through modality-aware labels.

### 4.2.3 Learning Temporal Heterogeneous Clues

We further induce the temporal difference in the attention model. Although the self-modality and cross-modality attention (Eqn. (4.2) and (4.3)) lead to a more comprehensive understanding by leveraging audio-visual contexts, however, we argue that these might harm the model performance since it obscures the temporal difference within an event video. It is necessary to introduce the temporal difference during the weakly-supervised training.

Since we do not have temporal annotation for each segment, we propose to leverage contrastive learning to alleviate the issue. Contrastive learning [11, 118] is popular in self-supervised learning. We design a proxy task that urges the attention model to pick the correct temporal segment from all distractor segments, which prevents the aggregated model from being dominated by a few segment features.

We use Contrastive Learning [32, 35, 120] to guide the aggregated representation $\hat{\mathbf{f}}^a_t$ to be close with the low-level visual feature $\mathbf{f}^v_t$ at the same timestamp, while is far away from visual features at other temporal segments. Thus, the positive sample is the original feature $\mathbf{f}^v_t$. As for the negative distractors, we use the visual features from the same video but from other temporal clips, *i.e.*, $\mathbf{f}^v_{t'}$, $t' \neq t$. The distractors can be regarded as hard examples for contrastive learning since the candidates are similar to the original clip representation $\mathbf{f}^v_t$.

With the positive target and these distractors, we can add auxiliary supervision to the model with contrastive learning,

$$\mathcal{L}_c = -\log \frac{\exp(\mathbf{f}^{v\intercal}_t \hat{\mathbf{f}}^a_t / \tau)}{\sum_j \exp(\mathbf{f}^{v\intercal}_j \hat{\mathbf{f}}^a_t / \tau)}, \tag{4.8}$$

where the features are L2-normalized, *i.e.*, $||\mathbf{f}_j^v|| = 1$, $||\hat{\mathbf{f}}_t^a|| = 1$. The temperature weight $\tau$ defines the sharpness of softmax function. Lower $\tau$ means a harder distribution. In our model we set its value to 0.2.

By using the binary cross-entropy loss together with the above contrastive loss, the attention model might not be dominated by some temporal segments. The aggregated feature would be more likely the information that happens at this segment instead of all context features, leading to better temporal localization performances.

## 4.3 Experiments

### 4.3.1 Experiment Settings

**The Look, Listen and Parse Dataset** [94] (LLP) contains 11849 videos with 25 event class. It contains a wide range of human activities and daily life videos. Each video is ten seconds long. For the *weakly-supervised* AVVP task, there are 10,000 videos for training, containing weak labels only. To evaluate AVVP performance, the 1,849 validation and test videos have fully annotated labels (dense annotations).

**Evaluation Metrics.** We evaluate our method at the segment level and the event level. F-scores are used to evaluate the predictions. The segment-level metrics measure segment-level event prediction accuracy. Besides segment-level performance, the event-level results are also reported to indicate the performance in real applications. We concatenate consecutive positive clips with the same event class, and obtain the event F-scores using 0.5 as the mIoU threshold. We also evaluate the overall performance by computing aggregated results, *i.e.*, "Type@AV" and "Event@AV". Specifically, Type@AV measures the mean event recognition accuracy. Event@AV is F-score by regarding sound and vision events for each example.

**Implementation Details.** We use both the ResNet-152 [36] model pre-trained using ImageNet and R(2+1)D [98] model pre-trained using Kinetics to extract visual

representations. We decode videos at 8 fps and input each segment (lasting one second) to obtain the 2D and 3D visual features. We regard the concatenation of the two features as the low-level vision feature. For the audio signals, we take the VGGish model [38] pre-trained using AudioSet [31] to extract 128-D features. The Adam optimizer is used to optimize the model with a learning rate of 0.0003. The batch size of 16. We change the learning rate to 0.00003 after 10 epochs. Our training pipeline includes three stages. First, we optimize a base model for audio-visual scene parsing using MIL and our proposed contrastive learning. Second, we freeze the model and evaluate each video by swapping its audio and visual tracks with other unrelated videos. At last, we re-train our MA from scratch with modality-aware labels. We name the final model as "**MA**" (Modality Aware) to distinguish it from the base model.

### 4.3.2 Comparison to State-of-the-art Methods

We compare our MA to weakly-supervised sound detection method TALNet [109], temporal action localization models STPN [69] and CMCS [59], and state-of-the-art audio-visual parsing methods including AVE [95], AVSDN [58], and HAN [94]. All the models, including ours, are trained for fair comparisons using the LLP training dataset only, including the same training data and pre-processed audio/visual features.

Table 4.1 shows the performance on the LLP test set. Our method beats the compared methods on all metrics. Specifically, on the audio-visual event prediction, our MA beats HAN [94] by 6.2 points (from 48.9% to 55.1%) at the segment level, and 6.0 points (from 43.0% to 49.0%) at the event level. The most significant improvement is found for visual event parsing, which validates our motivation that previous methods are suffered from the ambiguous overall labels of invisible events. The comparison demonstrates that our MA can predict significantly better event

| Event type | Models | Segment-level | Event-level |
|---|---|---|---|
| Audio-visual | AVE [95] | 35.4 | 31.6 |
| | AVSDN [58] | 37.1 | 26.5 |
| | HAN [94] | 48.9 | 43.0 |
| | **MA (Ours)** | **55.1** (+6.2) | **49.0** (+6.0) |
| Audio | TALNet [109] | 50.0 | 41.7 |
| | AVE [95] | 47.2 | 40.4 |
| | AVSDN [58] | 47.8 | 34.1 |
| | HAN [94] | 60.1 | 51.3 |
| | **MA (Ours)** | **60.3** (+0.2) | **53.6** (+2.3) |
| Visual | STPN [69] | 46.5 | 41.5 |
| | CMCS [59] | 48.1 | 45.1 |
| | AVE [95] | 37.1 | 34.7 |
| | AVSDN [58] | 52.0 | 46.3 |
| | HAN [94] | 52.9 | 48.9 |
| | **MA (Ours)** | **60.0** (+7.1) | **56.4** (+7.5) |
| Type@AV | AVE [95] | 39.9 | 35.5 |
| | AVSDN [58] | 45.7 | 35.6 |
| | HAN [94] | 54.0 | 47.7 |
| | **MA (Ours)** | **58.9** (+4.9) | **53.0** (+5.3) |
| Event@AV | AVE [95] | 41.6 | 36.5 |
| | AVSDN [58] | 50.8 | 37.7 |
| | HAN [94] | 55.4 | 48.0 |
| | **MA (Ours)** | **57.9** (+2.5) | **50.6** (+2.6) |

Table 4.1 : Results of the audio-visual video parsing task on the LLP test dataset.

categories with accurate temporal locations.

### 4.3.3 Ablation Studies

**Effectiveness of Modality-aware Refinement.** As shown in Table 4.2, "Baseline + R" is the model trained with modality-aware refinement. By leveraging clues between the audio and visual tracks and assigning different labels for the two modalities, we find the model performance gets significantly improved. Table 4.2 shows our model "Baseline + R" outperforms the baseline by about 4 points at audio-visual

| Event type | Models | Segment-level | Event-level |
|---|---|---|---|
| Audio-visual | Baseline | 48.9 | 43.0 |
| | Baseline + C | 49.7 | 43.8 |
| | Baseline + R | 52.6 | 45.8 |
| | Baseline + C + R | **55.1** | **49.0** |
| Audio | Baseline | 60.1 | 51.3 |
| | Baseline + C | **61.9** | 52.8 |
| | Baseline + R | 59.8 | 52.1 |
| | Baseline + C + R | 60.3 | **53.6** |
| Visual | Baseline | 52.9 | 48.9 |
| | Baseline + C | 53.1 | 49.4 |
| | Baseline + R | 57.5 | 54.4 |
| | Baseline + C + R | **60.0** | **56.4** |
| Type@AV | Baseline | 54.0 | 47.7 |
| | Baseline + C | 54.9 | 48.7 |
| | Baseline + R | 56.6 | 50.8 |
| | Baseline + C + R | **58.9** | **53.0** |
| Event@AV | Baseline | 55.4 | 48.0 |
| | Baseline + C | 56.2 | 49.0 |
| | Baseline + R | 56.6 | 49.4 |
| | Baseline + C + R | **57.9** | **50.6** |

Table 4.2 : Ablation studies on the LLP test dataset. "C" denotes the proposed contrastive learning for temporal localization. "R" is our modality-aware refinement by exchanging audio and visual channels.

event parsing evaluation metrics. Specifically, for the visual event parsing, the model with the modality-aware refinement significantly improves the performance by 4.6 points (from 52.9% to 57.5%) at the segment-level prediction and 5.5 points (from 48.9% to 54.4%) at the event level. It validates that ambiguous video-level labels harm model training since some events only appears in one modality.

**Analysis of Modality Bias in Refinement.** We further uncover the effect of modality-aware refinement by looking into modalities. We conduct experiments in-

cluding 1) only refining audio labels, 2) only refining visual labels, and 3) refining both modalities labels. The numbers are reported in Table 4.3. The most significant improvement is brought by refining event labels for visual parsing prediction. By refining visual parsing labels, we significantly improve the performance on segment-level visual parsing evaluation. The reason is that the visual content could only be captured for specific camera views, whether the object of interest might usually be outside of the video view. In contrast, the audio signals are collected by microphones, which are able to perceive all the event information of the scenes. Therefore, unmatched event labels are more common for visual modalities. By refining visual event labels for these *audible but not visible* videos, we observe a noticeable performance improvement on all the evaluation metrics except audio-only parsing.

Besides, we achieve further performance improvement by refining event labels for both modalities. Compared to "visual-only", the model trained with both modality refinement obtain considerable performance gain on all evaluation metrics.

**Effectiveness of Cross-modal Contrastive Learning.** Table 4.2 also shows the relative improvement brought by the cross-modal contrastive learning. Compared to the baseline, our model with the contrastive learning only ("Baseline + C") shows an improvement on audio-visual even parsing. The relative improvement is even more significant when combining with the modality-aware refinement. By comparing the model "Baseline + C + R" and model "Baseline + R", we can find the contrastive learning further improve the event parsing performance by about 2 points on most evaluation metrics. It indicates our proposed contrastive learning could introduce essential temporal differences for audio-visual video parsing.

**Analysis of different $\tau$ values.** We validate different $\tau$ values used in our MA. Table 4.4 shows the comparison of the segment-level audio-visual video parsing evaluation. Smaller $\tau$ leads to a sharper distribution. In experiments, we find the per-

| Modality | Aud | Vis | Aud-Vis | Type@AV | Event@AV |
|---|---|---|---|---|---|
| Audio only | 60.5 | 52.7 | 51.8 | 55.0 | 54.2 |
| Visual only | 60.4 | 59.0 | 53.5 | 57.9 | 57.1 |
| Both | 60.3 | 60.0 | 55.1 | 58.9 | 57.9 |

Table 4.3 : Analysis of the modality-aware refinement. "Audio" and "Visual" indicate that we only refine labels for the audio modality and the visual modality, respectively. Segment-level metrics are reported.

| $\tau$ | Aud | Vis | Aud-Vis | Type@AV | Event@AV |
|---|---|---|---|---|---|
| 0.1 | 61.3 | 58.3 | 54.5 | 58.4 | 57.8 |
| 0.2 | 60.3 | 60.0 | 55.1 | 58.9 | 57.9 |
| 0.3 | 60.5 | 60.3 | 54.9 | 58.7 | 57.9 |
| 0.4 | 60.3 | 59.9 | 55.0 | 58.5 | 57.3 |

Table 4.4 : Analysis on different $\tau$ values used in contrastive learning (Eqn (4.8)). Smaller $\tau$ leads to sharper probability distribution. Segment-level metrics are reported.

formances get slightly higher as $\tau$ decreases. Overall speaking, our model is not sensitive to the values of $\tau$ used in the contrastive learning (Eqn.(4.8)). In all other experiments, we set $\tau$ to 0.2.

### 4.3.4 Qualitative Results

We visualize the audio-visual video parsing results in Fig. 3.5. "Pred" shows the prediction from our models. "GT" is the ground truth annotation. Overall speaking, our model could correctly recognize the events happening in the video. But it makes mistakes on the temporal location of these events. For example, our

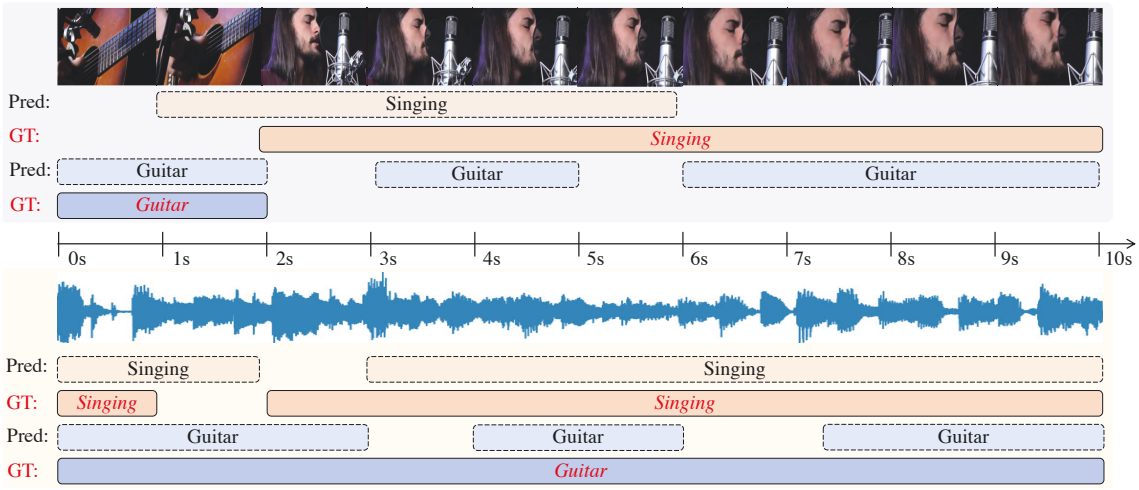


Figure 4.3 : Visualization results on the LLP test set. The upper and bottom figure shows visual and audio event parsing, respectively. "Pred" is the prediction result from our model, while "GT" indicates the ground truth annotation.

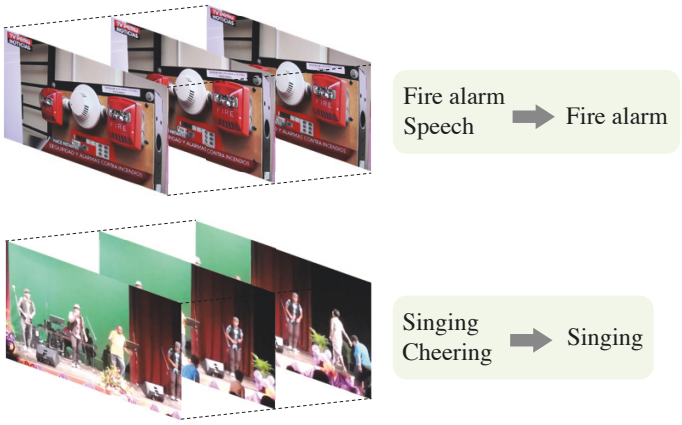


Figure 4.4 : Examples of our refined labels for the visual modality.

model still predicts guitar for the visual event parsing after 2s, although we could not find such clues of the guitar in the corresponding visual frames. The reason might be that the context feature aggregation collects too much information from the audio and video of other time stamps. For example, the audio clearly indicates "guitar" at this moment. Compared to the visual parsing, the audio event parsing

prediction is more reliable in general. The reason might be that audio is more clear and easy to be distinguished compared to complex visual frames.

We also show two examples of our modality-aware label refinement in Fig. 4.4. By exchanging audio and visual tracks among training videos, we localized event clues and found some events do not exist in the visual/audio modality. The upper case in the figure is a news video about the fire alarm event. Although the event labels are "fire alarm" and "speech" for the entire video in training, the model does not predict the "speech" event given the assembled video with exchanged channels (consisting of the original visual content and a new audio track). Through exchanging audio and visual signals, we could obtain a more accurate event label for the visual modality, *i.e.*, "fire alarm" only. In this way, we protect the visual model from being misled by the ambiguous overall event label "Speech".

## 4.4 Summary

This chapter focuses on the weakly-supervised audio-visual video parsing task, which predicts the audible or visible event categories and their temporal locations. We believe it harms the model training if we train both audio and visual models using the same overall labels. We propose to generate modality-aware event labels by swapping audio and visual tracks with other unrelated videos. If the newly assembled data predictions are not confident at the target event, there might be no event clues in the original visual/audio tracks. In this way, we could protect our models from being misled by ambiguous event labels. Besides, we further leverage heterogeneous clues temporally and induce temporal difference within videos by audio-visual contrastive learning. In conclusion, we found it useful by mining detailed annotations for different modalities. The inducing temporal difference also improves performance in the weakly-supervised AVVP task.

# Chapter 5

# Learning to Anticipate Egocentric Actions by Imagination

Anticipating the future action is an essential application in the video understating field. In this chapter, I introduce the ImagineRNN by generating intermediate future features on multi-modal features.

## 5.1 Introduction

Predicting the future video action has been very popular in recent research [10, 54]. The task has a lot of real applications if the system should react prior to an action getting executed. For example, in the autonomous driving system, the model should be able to predict if a vehicle will stop or the person will come across the road, since it should have some time for the power system to react before the accident.

Thus in this chapter, we investigate the action anticipation task. Given an observed video sequence, the model is supposed to predict the future action which will occur at $T$ seconds later, where $T$ is the anticipation time. Existing works [66, 20] first summarize those video content in the observation region, and then predict the future action based on current observations in a direct manner. However, these works neglect the time gap from the past to the future. It would benefit the action anticipating task if the model could find some clues to generate the missing frames in the unobserved period.

Learning by dreaming has been proven effecitve in active learning [105] and robot plicies [76]. In this chapter, we propose to tackle this issue by imagining

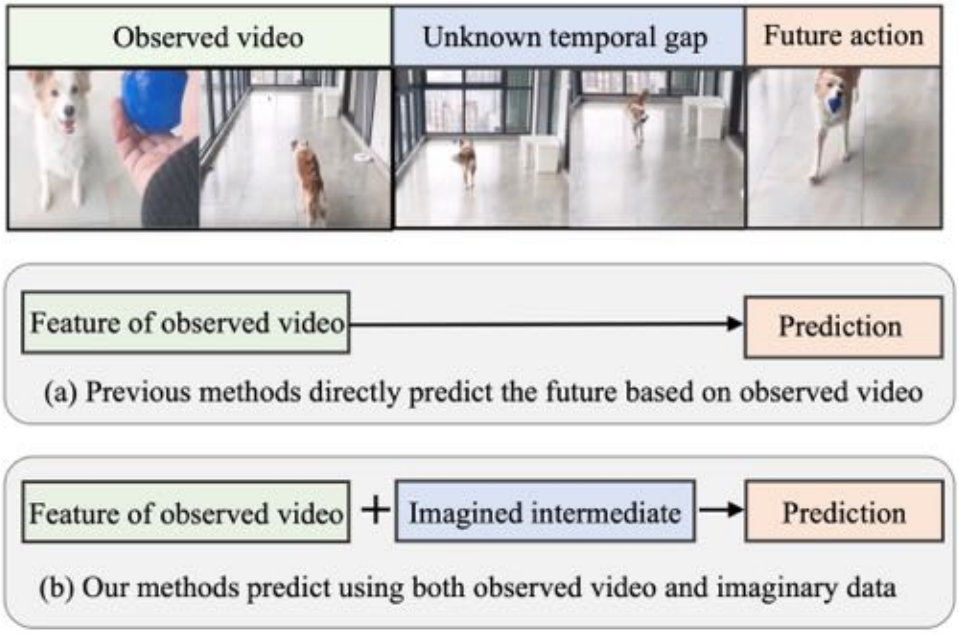


Figure 5.1 : We believe anticipating the middle representations improves the final action anticipation target.

the near future. First, we decompose the long-time action anticipation into a series of future feature predictions. We imagine how the visual feature changes in the very near future and then predict the future action labels based on these imagined representations. Specifically, we design the ImagineRNN to predict the next visual representation based on past observations in a step-wise manner. Since our target is to predict future action, it is unnecessary to waste model capacity on resolving the stochasticity of frame appearance changes due to camera motion and shadows in egocentric videos. Thus in ImagineRNN, we only generate the visual representation instead of raw pixels. The final anticipation is built on both the observed content and visual representation that we imagined within the anticipation time $T$.

Recently, some works [60, 63, 84, 21, 80] also propose to generate intermediate future frames or future content features using RNN or GAN architectures. Most of these works use regression loss functions (*e.g.*, $l_2$ loss or cosine loss) or discriminator

(justifying real or fake) to optimize their generator model. However, these optimization methods are too deterministic in training the generator model. There are only positive targets in these loss functions, leading to biased or sub-optimal optimization on the predicted future features. In addition, since actions are changed very quickly in egocentric videos, the predicted future states should be distinguishable in time sequences. Optimization with only positive targets would overlook the state changes in the future time period.

Our ImagineRNN differs from existing works in two aspects. First is that our ImagineRNN is optimized in the contrastive learning manner instead of feature regression. We propose a proxy task to train the ImagineRNN by selecting the correct future states from distractors. For the predicted future feature, we first build a set of candidates containing both the positive target (the ground truth future feature) and negative distractors (features from other time steps). Then we encourage the model to learn to identify the correct future state from candidates given the observed context. In this way, our ImagineRNN could essentially learn the change of future features. We found the new optimization method significantly improves the generalisability on the unseen test set.

Second, we further improve ImagineRNN by residual anticipation, *i.e.*, changing its target to predicting the feature difference of adjacent frames, instead of the entire frame feature. Different from [21, 80] that predict the entire optical flow frames or dynamic image, we only predict the feature changes between adjacent frames. The motivation is in three-folds. First, the difference between adjacent frame features is more important for forecasting the future. Predicting the video difference promotes the network to focus on the change of intermediate features, leading to better results on the future action anticipation. Second, it reduces the load of the ImagineRNN and thus saves the model capacity. In this way, the information the ImagineRNN has to predict is minimized, while the unchanged feature channels are directly carried

forward. Third, the unchanged content plays a role of shortcut connection, avoiding noise accumulation and gradient vanishing. To the best of our knowledge, we are the first to forecast the difference of frames in generating future features.

We conduct extensive experiments on two large-scale egocentric video datasets EPIC-KITCHENS [13] and EGTEA Gaze+ [52]. Results from the leaderboard of the EPIC-KITCHENS action anticipation challenge clearly show our model beats other existing single models.

To summarize, our contributions are summarized as follows: We propose ImagineRNN that breaks down the long-time action anticipation into a series of step-wise feature predictions of short periods, and then predicts the future action labels upon these imagined features. We reformulate the future feature prediction problem, and propose to optimize the ImagineRNN by picking the correct future states from lots of distractors, which essentially learns the change of future features compared to the traditional regression loss functions. We further replace the ImagineRNN's target by predicting the difference between adjacent frames, which helps the model focus on the feature change along time, leading to better anticipation performance. Experiments with different architectures validate the effectiveness of this change.

## 5.2 Proposed Approach

### 5.2.1 Egocentric Action Anticipation

**Task definition.** In the EPIC-Kitchens anticipation challenge [13], the egocentric action anticipation task is defined to predict the future action one second before it happens. In a more general task definition [20], the video is input in an online fashion, with a short video snippet consumed every $\alpha$ seconds, *i.e.*, the video is divided into segments of length $\alpha$. For an action occurring at time $\tau_s$, the model should anticipate the action by observing the video frames before $\tau_s - T$. In our

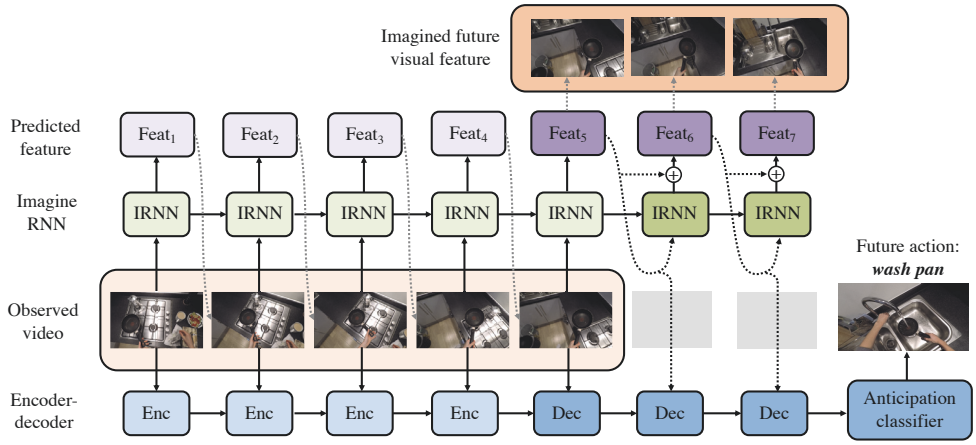


Figure 5.2 : The framework of our method. ImagineRNN predicts the next visual representation based on past observations in a step-wise manner. The imaginary features are input to the decoder to improve the anticipation performance. We propose to better optimize the ImagineRNN with the contrastive learning task. We further improve the ImagineRNN by forecasting the feature difference between frames, instead of generating the entire frame representations.

framework, our model is allowed to observe the video segment of length $(l - T)$ starting at time $(\tau_s - l)$ and ending at time $(\tau_s - T)$. Following [20], we use the same task setting and set $l = 3.5s$ and $\alpha = 0.25s$. We also validate our model under different anticipate time, *i.e.*, $T \in \{1.75s, 1.5s, 1.25s, 1s, 0.75s, 0.5s, 0.25s\}$. Note that it is more general compared to the task defined in [13], which only validates the model under anticipate time $T = 1$.

**CNN pre-training.** The input of our model is the frame-level feature provided by the pre-trained Temporal Segment Networks (TSN) [107] model. In action anticipation, the anticipation targets (objects and actions) do not always appear in the input video, making it hard to learn good representations for CNN models in an end-to-end manner. To avoid over-fitting and make the CNN model more meaningful, we follow [20] and pre-train the TSN model on the action recognition task.

Then the pre-trained CNN weights are fixed during the following training on our action anticipation task. We pre-process the videos and obtain different modalities features by pre-trained CNN models, *i.e.*, RGB frame features, optical flow frame features, and the object features.

**Encoder.** We take a Long Short-Term Memory (LSTM) [39] model as the temporal encoder. At each time step, the encoder takes as input the visual content that is being observed. Specifically, at each time-step $t$, we use the pre-trained TSN model to get the current frame feature $\mathbf{f}_t$. Then we input the feature $\mathbf{f}_t$ to update the memory. The new encoding hidden state $\mathbf{h}^E_{t+1}$ is obtained by updating the LSTM unit as follows:

$$\mathbf{h}^E_{t+1} = \texttt{Encoder}(\mathbf{f}_t, \mathbf{h}^E_t), \tag{5.1}$$

where $\mathbf{h}^E_t$ is the hidden state from the previous forward. We initialize the hidden state as zeros. To save memory and avoid noises, we only input the frames several seconds before the action occurring time $\tau_s$. Following [20], we take the frames from $(\tau_s - 4)$s to $(\tau_s - 2.5)$s as the input for the encoder.

**Decoder.** The decoder is an LSTM model that performs anticipation. It takes the observed information extracted from the EncodingRNN as the initial hidden states, and then recurrently takes the last observed frame as input. Based on the last output of the DecodingRNN, we use a fully connected layer as the classifier for the action anticipation prediction.

### 5.2.2 Bridging the gap between past and future

In the egocentric action anticipation task, it is hard to train a meaningful model due to the clear gap between past observations and future action. We alleviate this issue by decomposing the long-time prediction into a series of short-term forecasts. Then we design ImagineRNN to fill in the gap by producing the future visual repre-

sentation. In this way, the long-time reasoning is simplified by predicting the action based on past observations and future imaginary data.

Specifically, we break down the $T$ seconds anticipation into several short-term anticipations with each lasting $\alpha$ seconds ($\alpha < T$). Given the visual feature $\mathbf{f}_t$ at time $t$, the ImagineRNN is designed to generate the future visual feature $\hat{\mathbf{f}}_{t+1}$ by,

$$\mathbf{h}^I_{t+1} = \texttt{ImagineRNN}(\mathbf{f}_t, \mathbf{h}^I_t), \tag{5.2}$$

$$\hat{\mathbf{f}}_{t+1} = \phi(\mathbf{h}^I_{t+1}), \tag{5.3}$$

where $\mathbf{h}^I_t$ is the hidden state of ImagineRNN at time step $t$. $\phi(\cdot)$ is a transformation layer that maps the hidden state space to the visual feature space. The generated visual feature $\hat{\mathbf{f}}_{t+1}$ is supposed to fill in the gap between the past and future. In the framework, we input the output of ImagineRNN to the decoder to predict future action. Thus the prediction of ImagineRNN should be consistent with the ground truth visual content. Next, we illustrate how we optimize the ImagineRNN model efficiently in the action anticipation framework.

### 5.2.3 Optimization of ImagineRNN

In egocentric videos, the action states usually change very quickly. Thus the predicted future from ImagineRNN should be substantially different along with the anticipation time. The commonly used regression loss functions, such as $l_2$ loss, can hardly optimize the ImagineRNN to perceive the changes of action states. Differently, we propose a more effective optimization for the ImagineRNN by introducing the contrastive learning task, where the model is asked to pick the correct future states from lots of distractors. We use Noise Contrastive Estimation (NCE) [32] to encourage the predicted future feature $\hat{\mathbf{f}}_{t+1}$ to be close to the ground truth future state $\hat{\mathbf{f}}_{t+1}$. Compared to the regression losses, NCE does not require to resolve the low-level stochasticity strictly. Specifically, for the imagined future feature $\hat{\mathbf{f}}_t$ at

time $t$, the only positive target is the ground truth feature $\mathbf{f}_t$. We then build a set of candidates as distractors for the ground truth feature $\mathbf{f}_t$ at time $t$.

**Distractors.** The distractors contain easy negatives and hard negatives. The easy negatives contain the frame features from the other videos instead of the target video. We use the frame-level features from the other videos in the same mini-batch as the easy negatives for simplicity in the calculation. These candidates are easy to distinguish since these frames usually look different from the current video.

The hard negatives contain the frames from the same video but at different time steps, $\mathbf{f}'_t$ where $t' \neq t$. These candidates are hard to distinguish since they are very close to the ground truth frame feature $\mathbf{f}_t$. Distinguishing the hard negatives encourages ImagineRNN to generate essential intermediate features and capture the change of a series of future states.

**Contrastive Learning.** With the positive targets and these distractors, we can take contrastive learning as a proxy task for better optimizing the ImagineRNN. Inspired by recent representation learning work [120, 114], we first calculate the cosine similarity between the predicted feature and the candidates, $\mathbf{v}_j^{\mathrm{T}}\hat{\mathbf{f}}_t$, where $\mathbf{v}_j$ denotes the $j$-th distractors. Here we enforce all vectors to be L2-normalized feature embeddings, *i.e.*, $||\mathbf{v}_j|| = 1$, $||\hat{\mathbf{f}}_t|| = 1$, and $||\mathbf{f}_t|| = 1$. Thus we have the following objective function at the time step $t$,

$$\mathcal{L}_c = -\log \frac{\exp(\mathbf{f}_t{}^{\mathrm{T}}\hat{\mathbf{f}}_t/\tau)}{\sum_j \exp(\mathbf{v}_j{}^{\mathrm{T}}\hat{\mathbf{f}}_t/\tau) + \exp(\mathbf{f}_t{}^{\mathrm{T}}\hat{\mathbf{f}}_t/\tau)}, \tag{5.4}$$

where $\tau$ is a temperature parameter that controls the concentration level of the distribution. Higher $\tau$ leads to a softer probability distribution. We set $\tau = 0.2$ in our experiments.

With Eqn. (4.8), we optimize the ImagineRNN with a cross-entropy loss (negative log-likelihood), instead of the commonly used regression loss functions. During optimization, the loss function encourages the predicted feature $\hat{\mathbf{f}}_t$ to be close to

ground truth target $\mathbf{f}_t$, and also pushes the predicted feature $\hat{\mathbf{f}}_t$ to be distinct from these distractors. Thus the trained ImagineRNN could catch the change of action states at different times, which is essential in action anticipation.

2) The *future intention*. In addition, following [84, 21, 80], we also take the future intentions as additional supervision. The future intention is the purpose of the currently observed actions (the next future action), which explains the visual changes that happen during the unseen temporal region $T$. The intuition behinds it is that the generated visual representation should also benefit the anticipation task. Specifically, we input the generated visual feature to the decoder for several time steps during the anticipation time period. The decoder's last hidden state is further input to the action classifier for recognizing the future action. Then we use the Cross-Entropy loss on the final action anticipation to optimize the ImagineRNN. Denote the Cross-Entropy loss of the classifier as $\mathcal{L}_f$, the final loss is the sum of the two losses,

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_f. \tag{5.5}$$

### 5.2.4 Forecasting the difference between frames

However, the visual features of adjacent frames would be close since the backgrounds in frames are the same. Directly predicting the visual feature of future frames might waste model capacity in generating the unchanged background information. In addition, ImagineRNN might not essentially learn the change during future frames. Thus we propose to improve ImagineRNN by explicitly force it to predict the feature difference of adjacent frames, instead of the entire frame feature.

Specifically, we optimize ImagineRNN by learning to produce the difference between the current visual feature and the next one. The output of ImagineRNN is to forecast future *changes* of the visual feature given the current observation. Thus

we change Eqn. (5.3) to be,

$$\hat{\mathbf{f}}_{t+1} = \phi(\mathbf{h}^I_{t+1}) + \mathbf{f}_t. \tag{5.6}$$

Since we are designed to predict a series of intermediate frame features before anticipating the future action, we repeatedly use Eqn. (5.6) to generate a series of future frame features in an auto-regressive way. Suppose frame $t$ to be the last observed frame, we can obtain the imagined feature $\hat{\mathbf{f}}_{t+n}$ of future frame $t+n$ by,

$$\hat{\mathbf{f}}_{t+n} = \phi(\mathbf{h}^I_{t+n}) + \phi(\mathbf{h}^I_{t+n-1}) + ... + \phi(\mathbf{h}^I_{t+1}) + \mathbf{f}_t. \tag{5.7}$$

As can be seen in the above equation, predicting the difference sets up a shortcut connection between step-wise reconstructions, which helps ease the optimization of ImagineRNN and avoids noise accumulation during the auto-regressive future feature generation in testing. In addition, Predicting the frame difference promotes the model to focus on the change of intermediate features, which might be the core of future action anticipation.

## 5.3 Experiments

We first discuss the experimental setups and then compare our method with the state-of-the-art methods on two large-scale egocentric action datasets, EPIC-Kitchens and EGTEA Gaze+. Ablation studies and qualitative results are provided to show the effectiveness of our method.

### 5.3.1 Experimental Settings

**Datasets.** We perform experiments on two large-scale datasets of egocentric videos: EPIC-Kitchens [13] and EGTEA Gaze+ [52]. **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 activities performed are non-scripted, which

| Methods | Verb | | Noun | | Action | |
|---|---|---|---|---|---|---|
| | Top-1 Acc | Top-5 Acc | Top-1 Acc | Top-5 Acc | Top-1 Acc | Top-5 Acc |
| 2SCNN [13] | 29.76 | 76.03 | 15.15 | 38.56 | 04.32 | 15.21 |
| ATSN [13] | 31.81 | 76.56 | 16.22 | 42.15 | 06.00 | 28.21 |
| ED [26] | 29.35 | 74.49 | 16.07 | 38.83 | 08.08 | 18.19 |
| MCE [19] | 27.92 | 73.59 | 16.09 | 39.32 | 10.76 | 25.28 |
| Transitional [66] | 30.74 | 76.21 | 16.47 | 42.72 | 09.74 | 25.44 |
| RULSTM [20] | 33.04 | 79.55 | 22.78 | 50.95 | 14.39 | 33.73 |
| **Ours** ($l_2$ loss) | 35.26 | 79.66 | 22.57 | 52.04 | **15.07** | 34.66 |
| **Ours** (Contrast) | **35.44** | **79.72** | **22.79** | **52.09** | 14.66 | **34.98** |

Table 5.1 : Egocentric action anticipation results on the **Seen (S1)** test set of the EPIC-KITCHENS Action Anticipation Challenge [13] with anticipation time $T = 1$ second. All values are reported as percentage (%).

| Methods | Verb | | Noun | | Action | |
|---|---|---|---|---|---|---|
| | Top-1 Acc | Top-5 Acc | Top-1 Acc | Top-5 Acc | Top-1 Acc | Top-5 Acc |
| 2SCNN [13] | 25.23 | 68.66 | 09.97 | 27.38 | 02.29 | 09.35 |
| ATSN [13] | 25.30 | 68.32 | 10.41 | 29.50 | 02.39 | 06.63 |
| ED [26] | 22.52 | 62.65 | 07.81 | 21.42 | 02.65 | 07.57 |
| MCE [19] | 21.27 | 63.33 | 09.90 | 25.50 | 05.57 | 15.71 |
| Transitional [66] | 28.37 | 69.96 | 12.43 | 32.20 | 07.24 | 19.29 |
| RULSTM [20] | 27.01 | 69.55 | 15.19 | 34.38 | 08.16 | 21.10 |
| **Ours** ($l_2$ loss) | 27.35 | 69.78 | 15.36 | 35.34 | 08.54 | 20.79 |
| **Ours** (Contrast) | **29.33** | **70.67** | **15.50** | **35.78** | **09.25** | **22.19** |

Table 5.2 : Egocentric action anticipation results on the **Unseen (S2)** test set of the EPIC-KITCHENS Action Anticipation Challenge [13] with anticipation time $T = 1$ second. All values are reported as percentage (%).

makes the dataset very challenging and close to real-world data. This dataset is densely annotated with timestamps for each action so that it is ready for the action anticipation task. Actions in the EPIC-Kitchens dataset is annotated in the format of (`verb`, `noun`) pairs. The dataset contains 39,596 action annotations, 125 verbs, and 352 nouns. We considered all unique (`verb`, `noun`) pairs in the public training set, thus obtaining 2,513 unique actions. We use the same split as [20] and split the public training set of EPIC-Kitchens (28,472 action segments) into training (23,493 segments) and validation (4,979 segments) sets. **EGTEA Gaze+** contains 19 verbs, 51 nouns and 106 unique actions. We report the average performance across the three official splits provided by the authors of the dataset.

**Evaluation Metrics.** Following [20], we use the Top-k accuracy to evaluate our method. Under this evaluation metric, the prediction is deemed to be correct if the ground truth action falls in the top-k predictions. This metric is appropriate due to the uncertainty of future predictions [19, 49]. Many possible actions can follow an observation. We use the Top-5 accuracy as a class-agnostic measure. We also report Mean Top-5 Recall [19] as a class-aware metric. Top-5 recall for a given class $c$ is defined as the fraction of samples of ground truth class $c$ for which the class $c$ is in the list of the top-5 anticipated actions. Mean Top-5 Recall averages Top-5 recall values over classes. In [19], Top-5 Recall is averaged over the provided list of many-shot verbs, nouns, and actions. Performances are evaluated for verb, noun, and action predictions. Following [20], in training the only targets are the action labels, and our model is optimized to predict the action prediction. In the testing, we obtain the predictions for verb and noun by the marginalization on action predictions.

**Implementation Details.** We use Pytorch [74] to implement our framework. For the pre-trained action recognition model, we use a BNInception CNN [42] with the TSN framework to train the action recognition model. After pre-training, we resize the frame to $456 \times 256$ pixels and input it into the CNN model. The output (1024-

dimensional vectors) of the last global average pooling layer is used as the frame-level feature. The encoder, decoder, and the ImagineRNN are all single-layer LSTMs with 1024 hidden units. We use Stochastic Gradient Descent (SGD) to train the framework with a mini-batch size of 128 and a learning rate of 0.01 and momentum equal to 0.9. We train 100 epochs and apply early stopping at each training stage the same as [20]. This is done by choosing the intermediate and final models' iterations, which obtain the best Top-5 action anticipation accuracy for the anticipation time $T = 1s$ on the validation set. Following [20], we use the RGB frames, optical flow frames, and the object detection features as input for our model. We first train the model with each modality individually and then obtain the final prediction by a late fusion of the three models' predictions. In the following experiments, for fair comparisons with RULSTM, our model takes all the three modalities as input if not specified.

### 5.3.2 Comparison to the state-of-the-art methods

**Compared Methods** We compare our method with state-of-the-art action anticipation methods: Deep Multimodal Regressor (DMR) [104], Anticipation Temporal Segment Network (ATSN) of [13], Anticipation Temporal Segment Network trained with verb-noun Marginal Cross-Entropy Loss (MCE) [19], and the Encoder-Decoder LSTM (ED) introduced in [26]. We also compare with the early action recognition methods to the problem of egocentric action anticipation: Feedback Network LSTM (FN) [14], and an LSTM trained using the Exponential Anticipation Loss [43] (EL). To compare with state-of-the-art action anticipation methods, we reproduced a vanilla version of Feature Mapping RNN [84] without the kernalised RBF. For a fair comparison, we first train models with the three input modalities, *i.e.*, RGB features, optical flow features, and object features. Then we obtain the final prediction by a late fusion of the three models. Very recently, RULSTM [20] is proposed by

| Methods | Top-5 Action Accuracy @ different $T$ | | | | | |
|---|---|---|---|---|---|---|
| | 1.5 | 1.25 | 1.0 | 0.75 | 0.5 | 0.25 |
| DMR [104] | / | / | 16.9 | / | / | / |
| ATSN [13] | / | / | 16.3 | / | / | / |
| MCE [19] | / | / | 26.1 | / | / | / |
| FMRNN [84] | / | / | 32.7 | / | / | / |
| ED [26] | 23.2 | 24.8 | 25.8 | 26.7 | 27.7 | 29.7 |
| FN [14] | 24.7 | 25.7 | 26.3 | 26.9 | 27.9 | 29.0 |
| EL [43] | 26.4 | 27.4 | 28.6 | 30.3 | 31.5 | 33.6 |
| RULSTM [20] | 32.2 | 33.4 | 35.3 | 36.3 | 37.3 | 39.0 |
| **Ours** | **32.5** | **33.6** | **35.6** | **36.7** | **38.5** | **39.4** |

Table 5.3 : Action anticipation results on the EPIC-KITCHENS validation set under different anticipation time $T$. The performance is measured by the top-5 accuracy of action anticipation.

combining two LSTM to anticipate actions from egocentric video, where one LSTM is used to summarize the past, and the other is used to predict future actions based on the past future. We compare our method under both the standard anticipation setting (anticipation time $T = 1s$) and a more general anticipation setting (with variant anticipation time).

**Results on the EPIC-KITCHENS test server.** We compare our method with the state-of-the-art methods on the test server of EPIC-KITCHENS. Table 5.1 and Table 5.2 report results obtained from the official EPIC-KITCHENS action anticipation challenge submission server. The official test server computes the performances on two test sets, *i.e.*, the "seen" test, which includes the same scenes appearing in the training set (S1) and the "unseen" test set (S2), with kitchens not appearing in the training set. On both test sets, our method outperforms all previously reported

results under all metrics. On the S1 (seen) test set (Table 5.1), our method outperforms the previous method RULSTM by 1.25% on the Top-5 Action accuracy. On the S2 (unseen) test set (Table 5.2) where the videos are captured in new environments, our method significantly improves RULSTM in all metrics on Verb, Noun, and Action prediction. Note that we use the same input features with RULSTM, thus the comparison with RULSTM is a fair comparison, and the performance improvements over RULSTM are all from our algorithm instead of better features. These results demonstrate our method is better at anticipating future action.

**Results with Different Anticipation Time $T$.** Our method can also be used to predict future action under different anticipation time. Since each time step $\alpha$ in our method is 0.25s, we can evaluate the future anticipation every 0.25s. We compare our method with the state-of-the-art methods under different anticipation time $T \in \{2\text{s}, 1.75\text{s}, 1.5\text{s}, 1.25\text{s}, 1\text{s}, 0.75\text{s}, 0.5\text{s}, 0.25\text{s}\}$. The results are shown in Table 5.3. Note that some methods [13, 19, 104] can anticipate actions only at a fixed anticipation time. We found the proposed method always outperforms the strong competitor RULSTM [20] under all anticipation time $T$. Note that the results are reported on the validation set, where the models are selected by choosing the best validation performance, as used by RULSTM [20]. As indicated in [20], the results on the test server are more important in evaluating compared to the validation results.

**Results on the EGTEA Gaze+ dataset.** We also conduct experiments on the EGTEA Gaze+ dataset. Table 5.5 reports Top-5 action accuracy scores on EGTEA Gaze+ under different anticipation times. We use the same input modalities as RULSTM. Our method outperforms the compared methods under different anticipation time $T$. We also found the relative improvement is smaller on the EGTEA Gaze+ dataset compared to that on the EPIC-KITCHENS dataset. It might be because the EGTEA Gaze+ is relatively small in scale. It only consists 106 actions,

| Modality | Method | Top-1 Acc | Top-5 Acc |
|---|---|---|---|
| RGB | RULSTM [20] | 13.05 | 30.83 |
| | Ours w/o intention | 13.23 | 31.39 |
| | Ours w/o diff | 12.97 | 30.61 |
| | Ours | **13.68** | **31.58** |
| Flow | RULSTM [20] | 08.77 | 21.42 |
| | Ours w/o intention | 08.81 | 21.89 |
| | Ours w/o diff | 08.51 | 21.68 |
| | Ours | **09.23** | **22.06** |
| Obj | RULSTM [20] | 10.04 | 29.89 |
| | Ours w/o intention | **10.76** | 30.05 |
| | Ours w/o diff | 10.62 | 30.12 |
| | Ours | 10.72 | **30.27** |
| Fusion | RULSTM [20] | 15.00 | 35.24 |
| | Ours w/o intention | 15.04 | 35.17 |
| | Ours w/o diff | 14.91 | 34.98 |
| | Ours | **15.23** | **35.38** |

Table 5.4 : Comparison of the anticipated action accuracies with different modalities on the validation set.

which is far less than the $2,513$ actions in EPIC-KITCHENS. Thus the anticipation on the EPIC-KITCHENS dataset is more challenging.

### 5.3.3 Ablation Studies

We conduct ablation studies to evaluate the effectiveness of the two components of our method.

**Effectiveness of ImagineRNN.** Without our proposed ImagineRNN, the model is the baseline RULSTM. From Table 5.1 and Table 5.2, we can see the results of our baseline model only achieve 33.73% in Top-5 accuracy on the seen (S1) test set and

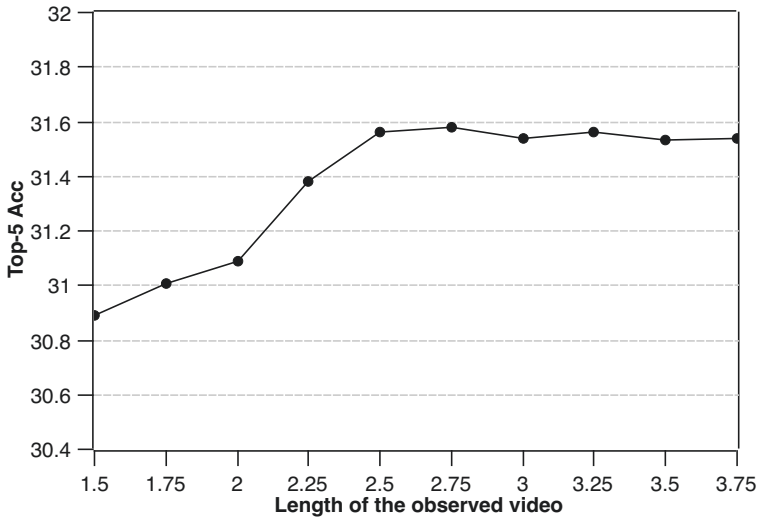


Figure 5.3 : Top-5 accuracies over different lengths of observed past for the encoder. The results are produced by our method with the RGB modality input.

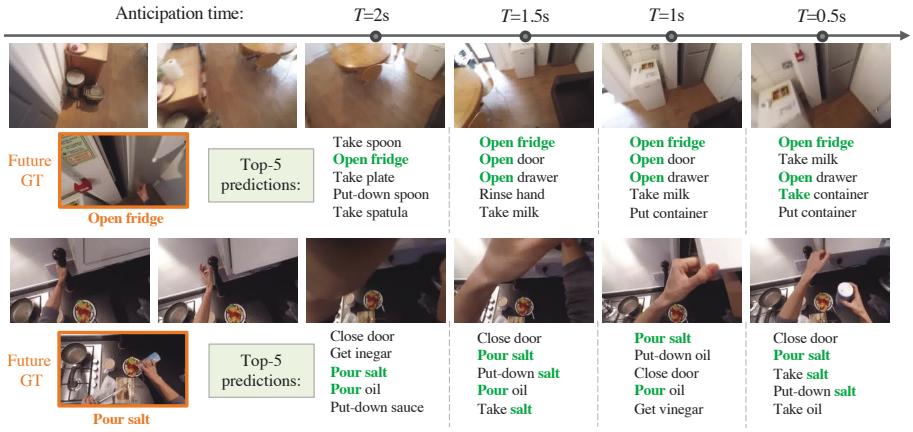


Figure 5.4 : Qualitative results with anticipation time $T = 2s, 1.5s, 1s, 0.5s$. From left to right, the observations are getting closer to future action. Orange indicates the ground truth, and green means our prediction matches the ground truth.

21.10% on the unseen (S2) test set. By adding our ImagineRNN to the framework, we observed a clear performance improvement on both test sets.

**Effectiveness of Contrastive Learning.** The common used optimization for the

| | Top-5 Action Accuracy% @ different $T$ | | | |
|---|---|---|---|---|
| | 1.0 | 0.75 | 0.5 | 0.25 |
| DMR [104] | 55.70 | / | / | / |
| ATSN [13] | 40.53 | / | / | / |
| MCE [19] | 56.29 | / | / | / |
| ED [26] | 50.22 | 51.86 | 49.99 | 49.17 |
| FN [14] | 60.12 | 62.03 | 63.96 | 66.45 |
| RL [64] | 62.56 | 64.65 | 67.35 | 70.42 |
| EL [43] | 64.62 | 66.89 | 69.60 | 72.38 |
| RULSTM [20] | 66.40 | 68.41 | 71.84 | 74.28 |
| Ours | **66.71** | **68.54** | **72.32** | **74.59** |

Table 5.5 : Anticipation results on the EGTEA Gaze+ dataset.

ImagineRNN is the regression loss functions ($l_2$ loss). In Table 5.1 and Table 5.2, we show the comparison of different optimization methods on the test set of EPIC-KITCHENS. Ours ($l_2$ loss) indicates our models optimized by the $l_2$ loss, while Ours (Contrastive) is the model optimized in the Contrastive Learning way, *i.e.*, picking the correct one from lots of distractors. With $l_2$ loss, our model achieves 34.66 on Top-5 Accuracy in the seen test set. In contrast, with Contrastive Learning, our method achieves 34.98% on Top-5 action accuracy. The improvement of Contrastive Learning is more clear in the unseen test set. With the proposed Contrastive Learning, the action anticipation result on the unseen set shows a 1.40% (22.19% versus 20.79%) improvement on the Top-5 action accuracy. The significant performance gap shows that contrastive learning is a better way to optimize ImagineRNN. It leads to a better generalisability across the various benchmarks.

**Effectiveness of Forecasting the Difference.** In Table 5.4, we show the comparison of results with and without forecasting the difference. We conduct ablation

studies on the RGB input, the flow input, and the fused modalities input. The results show a steady improvement by introducing to forecast the difference. Specifically, our method significantly outperforms the one (w/o diff) by 0.9 points on the Top-5 action accuracy on the RGB modality. Similarly, we found our approach also suppresses the model (w/o diff) with optical flow data as inputs. These comparison results prove the effectiveness of forecasting the difference instead of directly generating the whole visual feature. We also validate the effectiveness of forecasting the difference with other architectures. We replace the basic architectures of our ImagineRNN and the encoder-decoder by Gated Recurrent Unit (GRU), instead of the previously used LSTM. The results are shown in Table 5.7. It can be seen that our predicting the feature difference of adjacent frames still performs better with the GRU-based architecture.

**Effectiveness of Future Intention.** We also show the comparison of results with and without the future intention optimization Eqn. (5.5) in Table 5.4. The ablation studies show a small improvement brought by future intention. Specifically, our final model outperforms the one without future intension on the RGB and flow modalities by about 0.4% in Top-1 accuracy and 0.2% in Top-5 accuracy.

**Ablation studies over different lengths of the past.** We show the results over different lengths of observed past in Figure 5.3. Note the anticipation time $T$ is 1s for all experiments. It can be seen from the figure that the performance is relatively low if the encoder period is too short (*i.e.*, less than 2.25 seconds). As the encoding period gets longer, we found the performance gets steady. Inputting more observed frames did not lead to further performance improvement if the encoder period is longer than 2.5 seconds. The reason might be that actions usually change quickly in egocentric videos. Too early frames do not have strong correlations with future action.

| Methods | Verb | | Noun | | Action | |
|---|---|---|---|---|---|---|
| | Top-1 | Top-5 | Top-1 | Top-5 | Top-1 | Top-5 |
| $l_2$ | 27.4 | 69.8 | 15.4 | 35.3 | 8.5 | 20.8 |
| Con. + $l_2$ + Adv. | 27.9 | 70.3 | 14.3 | 34.7 | 8.5 | 20.7 |
| Con. + $l_2$ | 28.4 | 70.0 | 15.1 | 34.9 | 9.0 | 21.1 |
| Con. | **29.3** | **70.7** | **15.5** | **35.8** | **9.3** | **22.2** |

Table 5.6 : Comparison of different optimizations on the Unseen (S2) test set of the EPIC-KITCHENS Action Anticipation Challenge. "Con." indicates the contrastive learning loss. "Adv." indicates the adversarial loss used in GAN [21].

| Methods | Verb | | Noun | | Action | |
|---|---|---|---|---|---|---|
| | Top-1 | Top-5 | Top-1 | Top-5 | Top-1 | Top-5 |
| Ours (GRU w/o diff) | 32.9 | 78.7 | 22.0 | 49.2 | 13.3 | 32.3 |
| Ours (GRU with diff) | **33.7** | **79.7** | **22.7** | **50.2** | **14.0** | **33.2** |

Table 5.7 : Ablation studies of predicting the feature difference between adjacent frames with GRU-based architecture on the EPIC-KITCHENS Action Anticipation validation set.

**Discussion on different optimization methods.** We evaluate the models with different optimization methods on the test set of the EPIC-KITCHENS Action Anticipation Challenge. The results are shown in Table 5.6, where "Con." indicates the contrastive learning loss, and "Adv." is the adversarial loss used in GAN [21]. It can be seen that a combination of contrastive loss and $l_2$ loss does not outperform the one with the contrastive learning only. Besides, we add the adversarial loss in the model training, where the discriminator is a three-layer MLP. According to the validation results, we set the weight of the adversarial loss to be 0.01 in the

overall loss function and the discriminator's learning rate to be $2 \times 10^{-6}$. As shown in Table 5.6, the model trained with the combination of the three loss functions performs worst among all candidates. Our model trained with contrastive learning performs best among all candidates. The reason might be that contrastive learning helps to learn the change of future features essentially, since it needs to distinguish the positive target from lots of distractors. (frame features at other times).

### 5.3.4 Qualitative Results

We show some qualitative results of our method in Fig 5.4. From left to right, the observations are getting closer to future action. The orange box and "GT" indicate the ground truth of the future action. We list the Top-5 action predictions of our results at the anticipation time $T \in \{2s, 1.5s, 1s, 0.5s\}$. Green indicates the prediction matches the ground truth. Taking the first row as an example, the anticipations become more and more accurate as time flows. It is consistent with our motivation that long-time modeling might involve lots of noise. It is interesting to see the model always predicts "Open fridges" when $T$ is less than 2 seconds, probably because the fridge shows up in the observations at $T = 1.5s$. The other action candidates, including "Take milk" and "Open drawer", are also likely to take place in the near future.

## 5.4 Summary

In this chapter, we decompose the action anticipation task into a series of future frame feature predictions. We first imagine how the future feature changes and then predict future action based on these imagined representations. We found that ImagineRNN optimized with contrastive learning is superior to the typical anticipation models. In addition, we further propose to improve ImagineRNN by predicting the feature difference of adjacent frames instead of the whole frame content. It

helps promote the model to focus on the change of future states and avoid the noise accumulation during the auto-regressive future feature generation. Extensive experimental results on different architectures validate the effectiveness of the proposed method.

In conclusion, we found it useful to decompose action anticipation into lots of intermediate predictions. Focusing on the future state transition by contrastive learning and predicting future frames' differences improves the quality of intermediate predictions, leading to better results on the final action anticipation task.

# Chapter 6

# Novel Object Captioning

Vision language is an important part of multi-modal perception. In this chapter, I introduce the generalized captioning model that is able to generate descriptions for both seen and unseen objects.

## 6.1 Introduction

Image captioning is an important task in vision and language research [45, 78, 103, 121]. It aims at automatically describing an image by natural language sentences or phrases. Recent encoder-decoder architectures have been successful in many captioning tasks [7, 15, 45, 65, 78, 103, 121], in which the Convolutional Neural Network (CNN) is usually used as the image encoder, and the decoder is usually a Recurrent Neural Network (RNN) to sequentially predict the next word given the previous words. The captaining networks need a large number of image-sentence paired data to train a meaningful model.

These captioning models fail in describing the *novel objects* which are unseen words in the paired training data. For example, as shown in Figure 6.1, the Long-term recurrent Convolutional Networks (LRCN) [15] model cannot correctly generate captions for the novel object "zebra". As a result, applying the model in a new domain where novel objects can be visually detected requires professional annotators to caption new images to generate paired training sentences. This is labor expensive and thus limits the applications of captioning models.

A few works have been proposed recently to address the novel object captioning

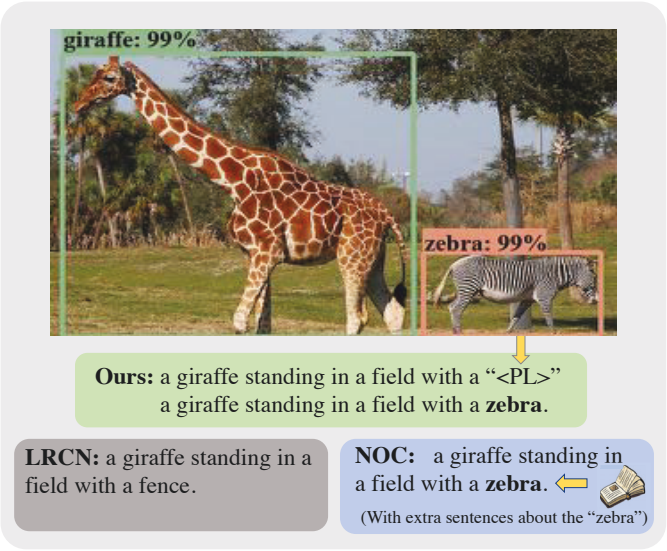


Figure 6.1 : An example of the novel object captioning. The colored bounding boxes show the object detection results. The novel object "zebra" is not present in the training data. We first generate the caption template with a placeholder "<PL>" that represents the novel object. We then fill in the placeholder with the word "zebra" from the object detection model.

problem [37, 100, 123, 17]. Essentially, these methods attempt to incorporate the class label produced by the pre-trained object recognition model. The class label can be used as the novel object description in language generation. To be specific, Henzdricks et al. [37] trained a captioning model by leveraging a pre-trained image tagger model and a pre-trained language sequence model from external text corpora. Yao et al. [123] exploited a pre-trained sequence model to copy the object detection result into the output sentence.

However, to feed the novel object description into the generated captions, existing approaches either employ the pre-trained language sequence model [37, 100] or require extra unpaired training sentences of the novel object [123]. In both cases, the *novel* objects have been used in training and, hence, is not really novel. A more precise meaning of *novel* in existing works is *unseen* in the *paired* training

sentences. However, the word is *seen* in the *unpaired* training sentences. For example, in Figure 6.1, existing methods require extra training sentences containing the word "zebra" to produce the caption, even though the object zebra is confidently detected in the image. The assumption that such training sentences of the novel object always exist may not hold in many real-world scenarios. It is considerably difficult to collect sentences about brand new products in a timely manner, *e.g.*, self-balancing scooters, robot vacuums, drones, and emerging topics trending on social media like fidget spinners. Someone may collect these training sentences about novel objects. However, it can still introduce language biases into the captioning model. For example, suppose training sentences are all about the bass (a sea fish). In that case, the captioning model will never learn to caption the instrument bass and may generate awkward sentences like "A man is eating a bass with a guitar amplifier."

This chapter tackles the image captioning for novel objects, where we do not need any training sentences containing the object. We utilize a pre-trained object detection model about the novel object. We call it *zero-shot* novel object captioning to distinguish it from the traditional problem setting [37, 100, 123]. In the traditional setting, in addition to the pre-trained object detection model, extra training sentences of the novel object are provided. In the zero-shot novel object captioning, there are *zero training sentences* about the novel object, *i.e.*, there is no information about the semantic meaning, sense, and context of the object. As a result, existing approaches of directly training the word embedding and sequence model become infeasible.

To address this problem, we propose a Decoupled Novel Object Captioner (DNOC) framework that is able to generate natural language descriptions without extra training sentences of the novel object. DNOC follows the standard encoder-decoder architecture but with an improved decoder. Specifically, we first design a sequence

model with the placeholder (SM-P) to generate captions with placeholders. The placeholder represents the unseen word for a novel object. Then we build a key-value object memory for each image, which contains the visual information and the corresponding words for objects. Finally, a query is generated to retrieve a value from the key-value object memory and the placeholder is filled by the corresponding word. In this way, the sequence model is fully decoupled from the novel object descriptions. Our DNOC is thus capable of dealing with the unseen novel object. For example, in Fig. 6.1, our method first generates the captioning sentence by generating a placeholder "<PL>" to represent any novel object. Then it learns to fill in the placeholder with "zebra" based on the visual object detection result.

## 6.2 The proposed Method

In this section, we first introduce the preliminaries and further show the two key parts of the proposed Decoupled Novel Object Captioner, *i.e.*, the sequence model with the placeholder (Section 6.2.2) and the key-value object memory (Section 6.2.3). The overview of the DNOC framework and training details are illustrated in Section 6.2.4 and Section 6.2.5, respectively .

### 6.2.1 Preliminaries

We first introduce the notations for image captioning. In image captioning, given an input image $I$, the goal is to generate an associated natural language sentence $\mathbf{s}$ of length $n_l$, denoted as $\mathbf{s} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{n_l})$. Each $\mathbf{w}$ represents a word and the length $n_l$ is usually varied for different sentences. Let $\mathcal{P} = \{(\mathbf{I}_1, \mathbf{s}_1), ..., (\mathbf{I}_{n_p}, \mathbf{s}_{n_p})\}$ be the set with $n_p$ image-sentences pairs. The vocabulary of $\mathcal{P}$ is $\mathcal{W}_{paired} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{N_t}\}$ which contains $N_t$ words. Each word $\mathbf{w}_i \in \{0, 1\}^{N_t}$ is a one-hot (1 of $N_t$) encoding vector. The one-hot vector is then embedded into a $D_w$-dimensional real-valued vector $\mathbf{x}_i = \phi_w(\mathbf{w}_i) \in \mathbb{R}^{D_w}$. The embedding function $\phi_w(\cdot)$ is usually a trainable lin-

ear transformation $\mathbf{x}_i = \mathbf{T}_w \mathbf{w}_i$, where $\mathbf{T}_w \in \mathbb{R}^{D_w \times N_t}$ is the embedding matrix. The typical architecture for captioning is the encoder-decoder model. In the followings, we show the encoding and the decoding procedures during the testing phase.

**The encoder.** We obtain the representation for an input image $\mathbf{I}$ by $\phi_e(\mathbf{I})$, where $\phi_e(\cdot)$ is the embedding function for encoder. The function $\phi_e$ is usually an ImageNet pre-trained CNN model with the classification layer removed. It extracts the top-layer outputs as the visual features.

**The decoder.** The decoder is a word-by-word sequence model designed to generate the sentence given the encoder outputs. In specific, at the first time step $t = 0$, a special token $\mathbf{w}_0$ ("<GO>") is the input to the sequence model, which indicates the start of the sentence. At time step $t$, the decoder generates a word $\mathbf{w}_t$ given the visual content $\phi_e(\mathbf{I})$ and previous words $(\mathbf{w}_0, ..., \mathbf{w}_{t-1})$. Therefore, we formulate the probability of generating the sentence $\mathbf{s}$ as

$$p(\mathbf{s}|\mathbf{I}) = \prod_{t=1}^{n_l} p(\mathbf{w}_t|\mathbf{w}_0, \ldots, \mathbf{w}_{t-1}, \phi_e(\mathbf{I})). \tag{6.1}$$

The Long Short-Term Memory (LSTM) [39] is commonly used as the decoder in visual captioning and natural language processing tasks [123, 37, 101]. The core of the LSTM model is the memory cell, which encodes the knowledge of the input that has been observed at every time step. There are three gates that modulate the memory cell updating, *i.e.*, the input gate, the forget gate, and the output gate. These three gates are all computed by the current input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$. The input gate controls how the current input should be added to the memory cell; the forget gate is used to control what the cell should forget from the previous memory; the output gate controls whether the current memory cell should be passed as output. Given inputs $\mathbf{x}_t$, $\mathbf{h}_{t-1}$, we get the predicted output word $\mathbf{o}_t$ by updating the LSTM unit at time step $t$ as follows:

$$\mathbf{o}_t, \mathbf{h}_t = \texttt{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}). \tag{6.2}$$

Combining Eqn (6.1) and Eqn (5.1), at $t$-th time step, the previous word $\mathbf{w}_{t-1}$ is taken as the decoder input $\mathbf{x}_t$. In the training stage, we feed the ground-truth word of the previous step as the model input. In the evaluation stage, we take the output $\mathbf{o}_{t-1}$ of the model at $(t-1)$-th time step as model input $\mathbf{x}_t$.

**Zero-Shot Novel Object Captioning.** This chapter studies the zero-shot novel object captioning task, where the model needs to caption novel objects *without* additional training sentence data about the object. The novel object words are shown neither in the paired image-sentence training data $\mathcal{P}$ nor unpaired sentence training data. We denote $\mathcal{W}_{unseen}$ as the vocabulary for the novel object words which are unseen in training. Given an input image $\mathbf{I}_n$ containing novel objects, the captioning model should generate a sentence with the corresponding unseen word $\tilde{\mathbf{w}} \in \mathcal{W}_{unseen}$ to describe the novel objects.

A notable challenge for this task is to deal with the out-of-vocabulary (OOV) words. The learned word embedding function $\phi_w$ is unable to encode the unseen words, since these words cannot simply be found in $\mathcal{W}_{paired}$. As a result, these unseen words cannot be fed into the decoder for caption generation. Previous works [37, 100, 123] circumvented this problem by learning the word embeddings of unseen words using additional sentences that contain the words. We denote these extra training sentences as $\mathcal{S}_{unpaired}$. Since the words of "novel" objects have been used in training, the "novel" objects are not really novel. However, in our zero-shot novel object task, we do not assume the availability of additional training sentences $\mathcal{S}_{unpaired}$ of the novel object. Therefore, we propose a novel approach to deal with the OOV words in the sequence model.

### 6.2.2 Sequence Model with the Placeholder

We propose the Sequence Model with the placeholder (SM-P) to fully decouple the sequence model from novel object descriptions. As discussed above, the classical

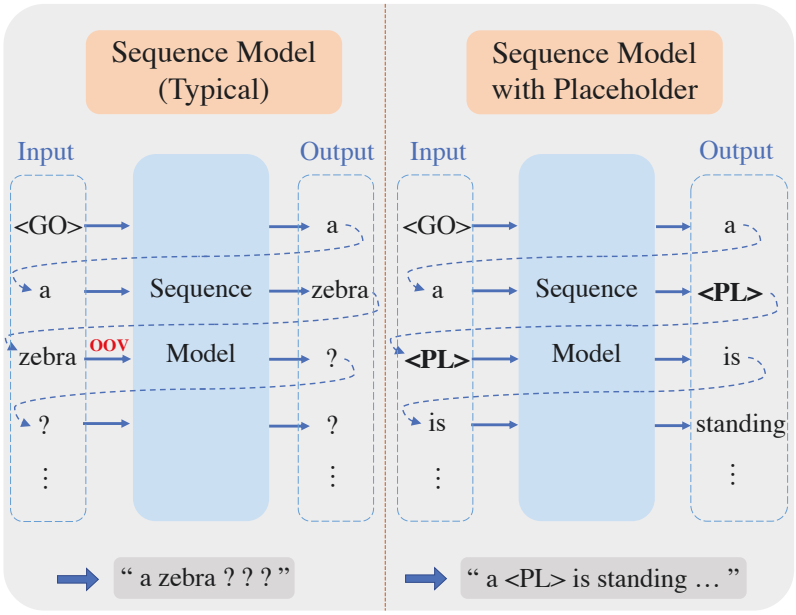


Figure 6.2 : The comparison of the typical sequence model and the proposed SM-P. In this example, "zebra" is an unseen word during training. The bottom are the sentences generated by the two models. The left is the classical sequence model, which cannot handle the input out-of-vocabulary (OOV) word "zebra". The right is our sequence model with the placeholder (SM-P). It generates the special token word "<PL>" (placeholder) to represent the novel object, and is able to continue to output the subsequent word given the input "<PL>".

sequence model cannot take an out-of-vocabulary word as input. To solve this problem, we design a new token, denoted as "<PL>". "<PL>" is the *placeholder* that represents any novel words $\tilde{\mathbf{w}} \in \mathcal{W}_{unseen}$. It is used in the decoder similarly to other tokens, such as "<GO >", "<PAD >", "<EOS >", "<UNKNOWN >" in most natural language processing works. We add the token "<PL>" into the paired vocabulary $\mathcal{W}_{paired}$ to learn the embedding. The training details for the placeholder are discussed in Section 6.2.5.

A new embedding $\phi_w$ is learned for "<PL>", which encodes all unknown words with a compact representation. The new representation could be jointly learned

with known words. We carefully designed the new token "<PL>" in both the input and the output of the decoder, which enables us to handle out-of-vocabulary words. When the decoder outputs "<PL>", our model utilizes the external knowledge from the object detection model to replace it with a novel description. Our SM-P is flexible that can be readily incorporated into the sequence to sequence model. Without loss of generality, we use the LSTM as the backbone of our SM-P. When the SM-P decides to generate a word about a novel object at time step $t$, it will output a special word $\mathbf{w}_t$, the token "<PL>" , as the output. At time step $t+1$, the SM-P takes the previous output word $\mathbf{w}_t$ "<PL>" as input instead of the novel word $\tilde{\mathbf{w}}$. In this way, regardless of the existence of unseen words, the word embedding function $\phi_w$ is able to encode all the input tokens. For example, in Figure 6.2, the classical sequence model cannot handle the out-of-vocabulary word "zebra" as input. Instead, the SM-P model outputs the "<PL>" token when it needs to generate a word about the novel object "zebra". This token is further fed to the decoder at the next time step. Thus, the subsequent words can be generated. Finally, the SM-P generates the sentence with the placeholder "A <PL>is standing ...". The "<PL>" token will be replaced by the novel word generated by the key-value object memory.

### 6.2.3 Key-Value Object Memory

To incorporate the novel words into the generated sentences with the placeholder, we exploit a pre-trained object detection model to build the key-value object memory.

A freely available object detection model is applied to the input images to find novel objects. For the $i$-th detected object $\texttt{obj}_i$, the CNN feature representations $\mathbf{f}_i \in \mathbb{R}^{1\times N_f}$ and the predicted semantic class label $\mathbf{l}_i \in \mathbb{R}^{1\times N_D}$ form a key-value pair, with the feature as key and the semantic label as the value. $N_f$ is the feature dimension of CNN representation, $N_D$ is the number of total detection classes. The

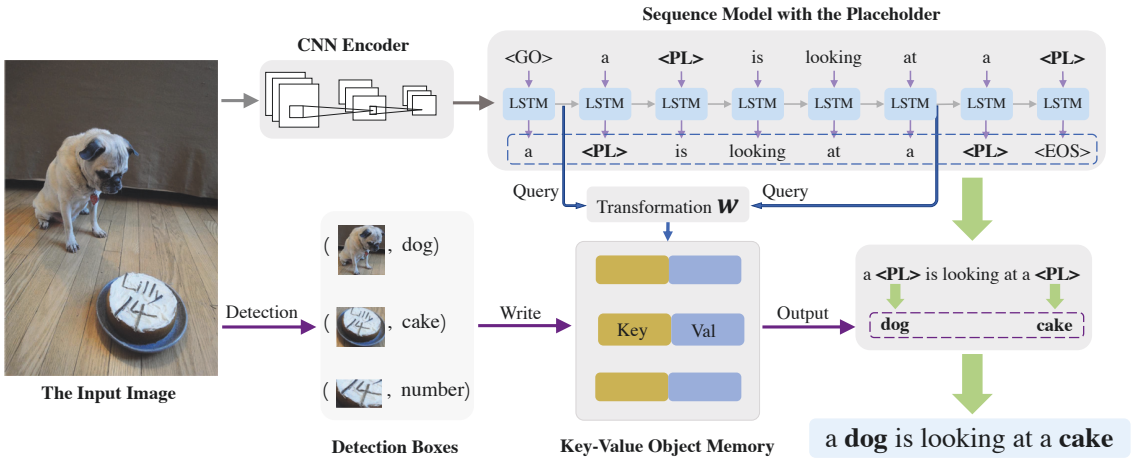


Figure 6.3 : The overview of the DNOC framework. We design a novel sequence model with the placeholder (SM-P) to handle the unseen objects by replacing them with the special token "<PL>". The SM-P first generates a sentence with placeholders, which refer to the unknown objects in an image. For example, in this figure, the "dog" and "cake" are unseen in the training set. The SM-P generates the sentence "a <PL>is looking at a <PL>". Meanwhile, we exploit a freely available object detection model to build a key-value object memory, which associates the semantic class labels (descriptions of the novel objects) with their appearance feature. When SM-P generates a placeholder, we take the linear transformation of the previous hidden state as a query to read the memory and output the correct object description, *e.g.*, "dog" and "cake". Finally, we replace the placeholders with the query results and generate the sentence with novel words.

key-value pairs associate the semantic class labels (descriptions of the novel objects) with their appearance feature. Following [44], we extract the CNN feature $\mathbf{f}_i$ for $\texttt{obj}_i$ from the ROI pooling layer of the detection model. Among all the detected results, the top $N_{det}$ key-value pairs are selected according to their confidence scores, which form the key-value object memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$. For each input image, the memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$ is initialized to be empty.

Let $\mathcal{W}_{det}$ be the vocabulary of the detection model, which consists of $N_D$ detection class labels. Note that each word in $\mathcal{W}_{det}$ is the detection class label in the *one-hot* format, since we cannot obtain trained word embedding function $\phi_w(\cdot)$ for the new word. To generate the novel word and replace the placeholder in the sentence at time step $t$, we define the query $\boldsymbol{q}_t$ to be a linear transformation of previous hidden state $\mathbf{h}_{t-1}$ when the model meets the special token "<PL>" at time step $t$:

$$\boldsymbol{q}_t = w\mathbf{h}_{t-1}, \tag{6.3}$$

where $\mathbf{h}_{t-1} \in \mathbb{R}^{N_h}$ is the previous hidden state at $(t-1)$-th step from the sequence model, and $w \in \mathbb{R}^{N_f \times N_h}$ is the linear transformation to convert the hidden state from semantic feature space to CNN appearance feature space. We have the following operations on the key-value memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$:

$$\boldsymbol{\mathcal{M}}_{\texttt{obj}} \leftarrow \texttt{WRITE}(\boldsymbol{\mathcal{M}}_{\texttt{obj}}, (\mathbf{f}_i, \mathbf{l}_i)), \tag{6.4}$$

$$\mathbf{w}_{\texttt{obj}} \quad \texttt{READ}(\boldsymbol{q}, \boldsymbol{\mathcal{M}}_{\texttt{obj}}). \tag{6.5}$$

- `WRITE` operation is to write the input key-value pair $(\mathbf{f}_i, \mathbf{l}_i)$ into the existing memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$. The input key-value pair is written to a new slot of the memory. The key-value object memory is similar to the support set widely used in many few-shot works [102, 81, 18]. The difference is that in their work the key-value memory is utilized for long-term memorization, while our motivation is to build a structured mapping from the detection bounding-box-level feature to its semantic label.

- `READ` operation takes the query $\boldsymbol{q}$ as input, and conducts content-based addressing on the object memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$. It aims to find related object information according to the similarity metric, $\boldsymbol{q}\boldsymbol{K}^T$. The output of `READ` operation is,

$$\mathbf{w}_{\texttt{obj}} = (\boldsymbol{q}\boldsymbol{K}^T)\boldsymbol{V}, \tag{6.6}$$

where $\boldsymbol{K}^T \in \mathbb{R}^{N_f \times N_{det}}, \boldsymbol{V} \in \mathbb{R}^{N_{det} \times N_D}$ are the vertical concatenations of all keys and values in the memory, respectively. The output $\mathbf{w}_{\texttt{obj}} \in \mathbb{R}^{N_D}$ is the combination of all semantic labels. In evaluation, the word with the max prediction is used as the query result.

### 6.2.4 Framework Overview

With the above two components, we propose the DNOC framework to caption images with novel objects. The framework is based on the encoder-decoder architecture with the SM-P and key-value object memory. For an input image with novel objects, we have the following steps to generate the captioning sentence:

We first exploit the SM-P to generate a captioning sentence with some placeholders. Each placeholder represents an unseen word/phrase for a novel object;

We then build a key-value object memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$ for each input based on the detection feature-label pairs $\{\mathbf{f}_i, \mathbf{l}_i\}$ on the image;

Finally, we replace the placeholders of the sentence with corresponding object descriptions. For the placeholder generated at time step $t$, we take the previous hidden state $\mathbf{h}_{t-1}$ from SM-P as a query to read the object memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$, and replace the placeholder by the query results $\mathbf{w}_{obj}$.

In the example shown in Figure 6.3, the "dog" and "cake" are the novel objects which are not present in training. The SM-P first generates a sentence "a <PL>is looking at a <PL>". Meanwhile, we build the key-value object memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$ based on the detection results, which contain both the visual information and the corresponding word (the detection class label). The hidden state at the step before each placeholder is used as the query to read from the memory. The memory will then return the correct object description, *i.e.*, "dog" and "cake". Finally, we replace the placeholders with the query results and thus generate the sentence with novel words "a dog is looking at a cake".

Note that we do not have a strict policy to force each PL matches a detected object. The selection of the detection class is based on the similarity/matching values by $\boldsymbol{q}\boldsymbol{K}^T$. During training the model has been trained to select the object that it needs in nature. However, using a strict policy such as Hungarian method may help improve this case.

### 6.2.5 Training

To learn how to exploit the "out-of-vocabulary" words, we modify the input and target for SM-P in training. We define $\mathcal{W}_{pd}$ as the intersection set of the vocabulary $\mathcal{W}_{paired}$ and vocabulary $\mathcal{W}_{det}$,

$$\mathcal{W}_{pd} = \mathcal{W}_{paired} \cap \mathcal{W}_{det}. \tag{6.7}$$

$\mathcal{W}_{pd}$ contains the words in the paired visual-sentence training data and the labels of the pre-trained detection model. For the $i$-th paired visual-sentence input $(\mathbf{I}_i, \mathbf{s}_i)$, we first encode the visual input by the encoder $\phi_e(\cdot)$. We modify the input annotation sentence $\mathbf{s}_i = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{n_l})$ of the sequence model SM-P by replacing each word $\mathbf{w}_i \in \mathcal{W}_{pd}$ with the token "<PL>". The new input word $\hat{\mathbf{w}}_t$ at $t$-th time step is,

$$\hat{\mathbf{w}}_t = \begin{cases} \langle PL \rangle, & \mathbf{w}_t \in \mathcal{W}_{pd} \\ \mathbf{w}_t, & otherwise. \end{cases} \tag{6.8}$$

We replace some known objects with the placeholder to help the SM-P learn to output the placeholder token at the correct place, and train the key-value object memory to output the correct word given the query. The actual input sentence for the SM-P is,

$$\hat{\mathbf{s}}_i = (\hat{\mathbf{w}}_0, \hat{\mathbf{w}}_1, ..., \hat{\mathbf{w}}_{n_l}). \tag{6.9}$$

We take the $\hat{\mathbf{s}}_i$ as the optimizing target for SM-P. Let $F_{SM}(\cdot)$ denote the function of SM-P, and $\boldsymbol{\theta}_{SM-P}$ denote its parameters. The output of function $F_{SM}(\cdot)$ is

the probability of next word prediction. Each step of word generation is a word classification on existing vocabulary $\mathcal{W}_{paired}$. SM-P is trained to predict the next word $\hat{\mathbf{w}}_t$ given $\phi_e(\mathbf{I})$ and sequence of words $(\hat{\mathbf{w}}_0, \hat{\mathbf{w}}_1, ..., \hat{\mathbf{w}}_{t-1})$. The optimizing loss for $\mathcal{L}_{SM-P}$ is,

$$\begin{aligned}&\mathcal{L}_{SM-P}(\hat{\mathbf{w}}_0, \hat{\mathbf{w}}_1, ..., \hat{\mathbf{w}}_{t-1}, \phi_e(\mathbf{I}); \boldsymbol{\theta}_{SM-P}) = \\ &-\sum_t \log(\texttt{softmax}_t(F_{SM}(\hat{\mathbf{w}}_0, \hat{\mathbf{w}}_1, ..., \hat{\mathbf{w}}_{t-1}, \phi_e(\mathbf{I}); \boldsymbol{\theta}_{SM-P}))),\end{aligned} \tag{6.10}$$

where the $\texttt{softmax}_t$ denotes the softmax operation on the $t$-th step.

For the key-value object memory $\boldsymbol{\mathcal{M}}_{\texttt{obj}}$, we define the optimizing loss by comparing the query result $\mathbf{w}_{obj_t}$ from object memory and the word $\mathbf{w}_t$ from annotation,

$$\mathcal{L}_{\boldsymbol{\mathcal{M}}_{\texttt{obj}}} = \sum_t a_t CE(\mathbf{w}_{obj_t}, \mathbf{w}_t), \tag{6.11}$$

where $CE(\cdot)$ is the cross-entropy loss function, and $a_t$ is the weight at time step $t$ that is calculated by,

$$a_t = \begin{cases} 1, & \mathbf{w}_t \in \mathcal{W}_{pd} \\ 0, & \text{otherwise.} \end{cases} \tag{6.12}$$

There are two trainable components in optimizing Eqn. (6.11). One is the query $\boldsymbol{q}$, the hidden state from the LSTM model. The other is the linear transformation on detection features in the computation of the memory key. We simultaneously minimize the two loss functions. The final objective function for the DNOC framework is,

$$\mathcal{L} = \mathcal{L}_{SM-P} + \mathcal{L}_{\boldsymbol{\mathcal{M}}_{\texttt{obj}}}. \tag{6.13}$$

## 6.3 Experiments

We first discuss the experimental setups and then compare DNOC with the state-of-the-art methods on the held-out MSCOCO dataset. Ablation studies and qualitative results are provided to show the effectiveness of DNOC.

### 6.3.1 The held-out MSCOCO dataset

The MSCOCO dataset [57] is a large-scale image captioning dataset. For each image, there are five human-annotated paired sentence descriptions. Following [37, 100, 123] , we employ the subset of the MSCOCO dataset, but excludes all image-sentence paired captioning annotations which describe at least one of eight MSCOCO objects. The eight objects are chosen by clustering the vectors from the word2vec embeddings over all the 80 objects in MSCOCO segmentation challenge [37]. It results in the final eight novel objects for evaluation, which are "bottle", "bus", "couch", "microwave", "pizza", "racket", "suitcase", and "zebra". These novel objects are held-out in the training split and appear only in the evaluation split. We use the same training, validation, and test split as in [37].

### 6.3.2 Experimental Settings

**The Object Detection Model**. We employ a freely available pre-trained object detection model to build the key-value object memory. Specifically, we use Faster R-CNN [79] model with Inception-ResNet-V2 [91] to generate detection bounding boxes and scores. The object detection model is pre-trained on all the MSCOCO training images of 80 objects, including the eight novel objects. We use the pre-trained models released by [41] which are publicly available. For each image, we write the top $N_{det} = 4$ detection results to the key-value object memory.

**Evaluation Metrics**. Metric for Evaluation of Translation with Explicit Ordering (METEOR) [6] is an effective machine translation metric that relies on the use of stemmers, WordNet [67] synonyms and paraphrase tables to identify matches between candidate sentence and reference sentences. However, as pointed in [37, 123, 100], the METEOR metric is not well designed for the novel object captioning task. It is possible to achieve high METEOR scores even without mentioning the novel objects. Therefore, to better evaluate the description quality, we also use

Table 6.1 : The comparison with the state-of-the-art methods on the eight novel objects in the held-out MSCOCO dataset. All F1-score values are reported as percentage (%).

| Settings | Methods | $F_{bottle}$ | $F_{bus}$ | $F_{couch}$ | $F_{microwave}$ | $F_{pizza}$ | $F_{racket}$ | $F_{suitcase}$ | $F_{zebra}$ | $F_{average}$ | METEOR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| With External Semantic Data | DCC [37] | 4.63 | 29.79 | 45.87 | 28.09 | 64.59 | 52.24 | 13.16 | 79.88 | 39.78 | 21 |
| | NOC [100] | | | | | | | | | | |
| | –(One hot) | 16.52 | 68.63 | 42.57 | 32.16 | 67.07 | 61.22 | 31.18 | 88.39 | 50.97 | 20.7 |
| | –(One hot+Glove) | 14.93 | 68.96 | 43.82 | 37.89 | 66.53 | 65.87 | 28.13 | 88.66 | 51.85 | 20.7 |
| | LSTM-C[123] | | | | | | | | | | |
| | –(One hot) | 29.07 | 64.38 | 26.01 | 26.04 | 75.57 | **66.54** | 55.54 | **92.03** | 54.40 | 22 |
| | NBT+G [62] | 7.1 | 73.7 | 34.4 | **61.9** | 59.9 | 20.2 | 42.3 | 88.5 | 48.5 | **22.8** |
| Zero-shot | LRCN [15] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19.33 |
| | DNOC w/o memory | 26.91 | 57.14 | 46.05 | 41.88 | 58.50 | 18.41 | 48.04 | 75.17 | 46.51 | 20.41 |
| | **DNOC (ours)** | **33.04** | **76.87** | **53.97** | 46.57 | **75.82** | 32.98 | **59.48** | 84.58 | **57.92** | 21.57 |

Table 6.2 : The comparison of our method and the baseline LRCN [15] on the six known objects in the held-out MSCOCO dataset. Per-object F1-scors and averaged F1-scores are reported as percentage (%).

| Methods | $F_{bear}$ | $F_{cat}$ | $F_{dog}$ | $F_{elephant}$ | $F_{horse}$ | $F_{motorcycle}$ | $F_{average}$ |
|---|---|---|---|---|---|---|---|
| LRCN [15] | 66.23 | 75.73 | 53.62 | 65.49 | 55.20 | 71.45 | 64.62 |
| DNOC | 62.86 | 87.28 | 71.57 | 77.46 | 71.20 | 77.59 | 74.66 |

the F1-score as an evaluation metric following [37, 123, 100]. F1-score considers false positives, false negatives, and true positives, indicating whether a generated sentence includes a new object.

**Implementation Details**. Following [123, 37, 100], we use a 16-layer VGG [89] pre-trained on the ImageNet ILSVRC12 dataset [68] as the visual encoder. The CNN encoder is fixed during model training. The decoder is an LSTM with cell size 1,024 and 15 sequence steps. For each input image, we take the output of the fc7 layer from the pre-trained VGG-16 model with 4,096 dimensions as the image representation. The representations are processed by a fully connected layer and then fed to the decoder SM-P as the initial state. For the word embedding,

unlike [123, 37], we do not exploit the per-trained word embeddings with additional knowledge data. Instead, we learn the word embedding $\phi_w$ with 1,024 dimensions for all words, including the placeholder token. We implement our DNOC model with TensorFlow [1]. Our DNOC is optimized by ADAM [47] with the learning rate of $1 \times 10^{-3}$. The weight decay is set to $5 \times 10^{-5}$. We train the DNOC for 50 epochs and choose the model with the best validation performance for testing.

### 6.3.3 Comparison to state-of-the-art results

We compare our DNOC with the following state-of-the-art methods on the held-out MSCOCO dataset.

(1). *Long-term Recurrent Convolutional Networks (LRCN)* [15]. LRCN is one of the basic RNN-based image captioning models. Since it has no mechanism to deal with novel objects, we train LRCN only on the paired visual-sentence data.

(2). *Deep Compositional Captioner (DCC)* [37]. DCC leverages a pre-trained image tagger model from large object recognition datasets and a pre-trained language sequence model from external text corpora. The captioning model is trained on the paired image-sentence data with the two pre-trained models.

(3). *Novel Object Captioner (NOC)* [100]. NOC improves the DCC to an end-to-end system by jointly training the visual classification model, language sequence model, and the captioning model.

(4). *LSTM-C* [123]. LSTM-C leverages a copying mechanism to copy the detection results to the output sentence with a pre-trained language sequence model.

(5). *Neural Baby Talk (NBT)* [62]. NBT incorporates visual concepts from object detectors to the sentence template. They manually define a category mapping list to replace the novel object's word embedding with an existing one to incorporate the novel words.

Table 6.1 summarizes the F1 scores and METEOR scores of the above methods and our DNOC on the held-out MSCOCO dataset. All the state-of-the-art methods except LRCN use additional semantic data containing the words of the eight novel objects. Nevertheless, without external sentence data, our method achieves competitive performance to state of the art. Our model, on average, yields a higher F1-score than the best state-of-the-art result (57.92% versus 55.66%). The improvement is significant, considering our model uses no additional training sentences. Our METEOR score is slightly worse than the LSTM-C with GloVe [123]. One possible reason is that much more training sentences containing the novel words are used to learn the LSTM-C model.

Table 6.2 shows the comparison of our DNOC and the baseline method LRCN on the known objects in the held-out MSCOCO dataset. Our method achieves much higher F1-scores on the known objects than LRCN, indicating that DNOC also benefits the captioning on the known objects. DNOC enhances the ability of the model to generate sentences with the objects shown in the image. The results strongly support the validity of the proposed model in both the novel objects and the known objects.

### 6.3.4 Ablation Studies

The ablation studies are designed to evaluate the effectiveness of each component in DNOC.

**The effectiveness of SM-P and key-value object memory.** We conduct the ablation studies on the held-out MSCOCO dataset. The results are shown in Table 6.3. The "DNOC w/o detection model" indicates the DNOC framework with SM-P but without any detection objects as input. All the objects in the visual inputs will not be detected. Thus, the placeholder token remains in the final generated sentence. We observe a performance drop compared to LRCN. "DNOC w/o object

Table 6.3 : Ablation studies in terms of Averaged F1-score and METEOR score on the held-out MSCOCO. "LRCN" is the baseline method. "DNOC w/o detection model" indicates the DNOC framework without any detected objects as input. "DNOC w/o object memory" indicates the DNOC framework with SM-P but without the key-value object memory.

| Model | $F1_{average}$ | METEOR |
|---|---|---|
| LRCN [15] | 0 | 19.33 |
| DNOC w/o detection model | 0 | 17.52 |
| DNOC w/o object memory | 46.51 | 20.41 |
| DNOC | 57.92 | 21.57 |

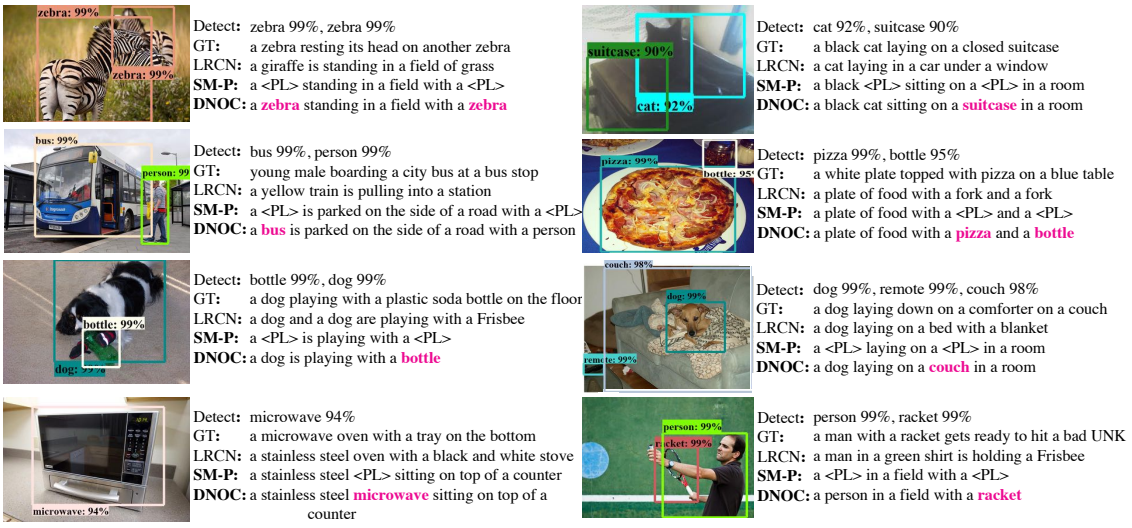


Figure 6.4 : Qualitative results for the held-out MSCOCO dataset. The words in pink are not present during training. "Detected" shows the object detection results. "GT" and "LRCN" are the human-annotated sentences and the sentences generated by LRCN, respectively. "SM-P" indicates the sentence generated by SM-P (the first step of DNOC). The SM-P first generates a sentence template with a placeholder, and DNOC further feeds the detection results into the placeholder.

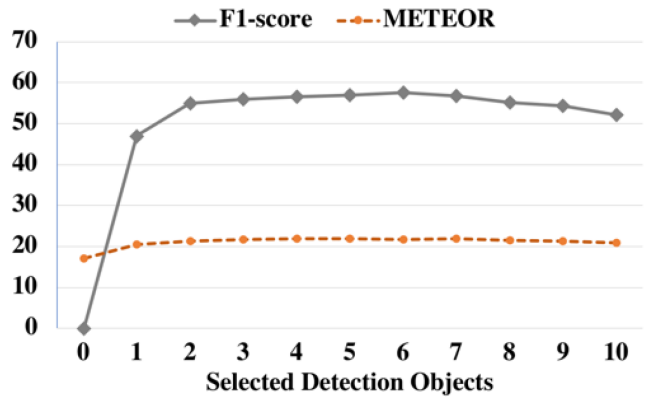


Figure 6.5 : The performance curve with different number of selected detected objects $N_{det}$.

memory" indicates the DNOC framework with SM-P but without the key-value object memory. In this experiment, we take the top $N_{det}$ confident detected labels and randomly feed them into the placeholder. Only with the SM-P component, "DNOC w/o object memory" outperforms LRCN by 46.51% in F1-score and 1.08% in METEOR score, which shows the effectiveness of the SM-P component. Our full DNOC framework outperforms "DNOC w/o object memory" by 11.41% in F1-score and 1.16% in METEOR score. It validates that the key-value object memory can enhance the semantic understanding of the visual content. From the experimental results, we can conclude that our full DNOC framework with SM-P and the key-value object memory greatly improves the performance in both the F1-score and the METEOR score. It shows that the two components are effective in exploiting the external detection knowledge.

**Analysis of the number of selected detection objects $N_{det}$.** $N_{det}$ is the number of selected top detection results. We show the performance curves with different $N_{det}$ values in Figure 6.5. When $N_{det}$ varies in a range from two to ten, we can see that the curves of F1-score and METEOR are relatively smooth. When we only adopt one detected object to build the object memory $\mathcal{M}_{\texttt{obj}}$, the F1-score

significantly drops. If we do not write any detection results into the memory, the F1-score is zero. This curve also demonstrates the effectiveness of the key-value object memory.

### 6.3.5 Qualitative Result

In Figure 6.4, we show some qualitative results on the held-out MSCOCO dataset. We take the "zebra" image in the first row as an example. The classic captioning model LRCN [15] could only describe the image with the wrong word "giraffe", where "zebra" does not exist in the vocabulary. SM-P first generates the sentence with the placeholder, where each placeholder represents a novel object. The DNOC then replaces the placeholders with the detection results by querying the key-value object memory. It generates the sentence with the novel word "zebra" in the correct place. As can be seen, zero-shot novel object captioning is a very challenging task since the evaluation examples contain unseen objects and no additional sentence data is available.

## 6.4 Summary

This chapter tackles the novel object captioning under a challenging condition where no sentence of the novel object is available. We propose a novel Decoupled Novel Object Captioner (DNOC) framework to generate natural language descriptions of the novel object. Our experiments validate its effectiveness on the held-out MSCOCO dataset. The comprehensive experimental results demonstrate that DNOC outperforms the state-of-the-art methods for captioning novel objects.

# Chapter 7

# Conclusion and Future Works

In this thesis, I investigated several works on multi-modal learning and video analysis with deep neural networks. Compared to single-modal perception models, the proposed cross-modal interaction and contrastive learning are proven to be very successful. The relation between modalities can be leveraged as the strong clues for overall video understanding. The intrinsic idea is that multi modalities contain different but consistent information reflecting the real world. I contributed to three tasks: audio-visual video understanding, multi-modal future action anticipation, and vision and language captioning.

For audio-visual video understanding, I developed two novel pipelines in Chapter 3 and Chapter 4, which study the synchronized and asynchronous audio-visual event parsing. The interaction between the audio and visual streams is very helpful in providing cross-modal supervision signals. A drawback of these two methods is that they only take the overall image feature as input. This would neglect some small objects in the visual channel. In addition, current audio-visual video understanding is more coarse-grained. In the future I might study the more fine-grained audio-visual video understanding such as visually localizing music beats in a dancing video. This is more challenging and requires better cross-modal association.

For multi-modal video action anticipation, I proposed an ImagineRNN to boost the anticipation performance by generating the intermediate features in the future gap. I found it useful to decompose action anticipation into lots of intermediate predictions. In future works, I might further explore the uncertainty of the future in

the egocentric action anticipation task, which is a limitation of the current work. In addition, transformer is a new architecture that has been proven successful in many multi-modal learning fields. Thus I might further study how to improve the current work by using the transformer design.

For novel object captioning, I developed a two-stage decoupled network by decomposing the sentence template generating and the visual object recognition. However, the current model only use a single token to represent all unseen objects in the current DNOC framework. It might lead to better performance if the model could leverage the association between visual graph (vision) and the semantic graph (language). It requires a precise understanding of both visual concepts and semantic concepts and their association. This is a long-term goal for my future study on the vision language field.

# Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[2] Y. Abu Farha, A. Richard, and J. Gall, "When will you do what?-anticipating temporal occurrences of activities," in *CVPR*, 2018.

[3] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *ICML*, 2013.

[4] R. Arandjelovic and A. Zisserman, "Look, listen and learn," *ICCV*, 2017.

[5] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *NIPS*, 2016.

[6] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *ACL-W*, 2005, pp. 65–72.

[7] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *NIPS*, 2015, pp. 1171–1179.

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

[9] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman, "Soundspaces: Audio-visual navigation in 3d environments," in *ECCV*, 2020.

[10] L. Chen, J. Lu, Z. Song, and J. Zhou, "Part-activated deep reinforcement learning for action prediction," in *ECCV*, 2018, pp. 421–436.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICLR*, 2020.

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[13] 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 *ECCV*, 2018.

[14] R. De Geest and T. Tuytelaars, "Modeling temporal structure with lstm for online action detection," in *WACV*, 2018.

[15] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015, pp. 2625–2634.

[16] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," in *SIGGRAPH*, 2018.

[17] Q. Feng, Y. Wu, H. Fan, C. Yan, M. Xu, and Y. Yang, "Cascaded revision network for novel object captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3413–3421, 2020.

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1126–1135.

[19] A. Furnari, S. Battiato, and G. M. Farinella, "Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation," in *ECCV-W*, 2018.

[20] A. Furnari and G. M. Farinella, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention." in *ICCV*, 2019.

[21] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Predicting the future: A jointly learnt model for action anticipation," in *ICCV*, 2019, pp. 5562–5571.

[22] C. Gan, D. Huang, P. Chen, J. B. Tenenbaum, and A. Torralba, "Foley music: Learning to generate music from videos," in *ECCV*, 2020.

[23] C. Gan, D. Huang, H. Zhao, J. B. Tenenbaum, and A. Torralba, "Music gesture for visual sound separation," in *CVPR*, 2020.

[24] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum, "Look, listen, and act: Towards audio-visual embodied navigation," in *ICRA*, 2020.

[25] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba, "Self-supervised moving vehicle tracking with stereo sound," in *ICCV*, 2019.

[26] J. Gao, Z. Yang, and R. Nevatia, "Red: Reinforced encoder-decoder networks for action anticipation," *BMVC*, 2017.

[27] R. Gao, R. Feris, and K. Grauman, "Learning to separate object sounds by watching unlabeled video," in *ECCV*, 2018.

[28] R. Gao and K. Grauman, "2.5 d visual sound," in *CVPR*, 2019.

[29] ——, "Co-separating sounds of visual objects," in *ICCV*, 2019.

[30] R. Gao, T.-H. Oh, K. Grauman, and L. Torresani, "Listen to look: Action recognition by previewing audio," in *CVPR*, 2020.

[31] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[32] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010.

[33] T. Han, W. Xie, and A. Zisserman, "Memory-augmented dense predictive coding for video representation learning," in *ECCV*, 2020.

[34] D. Harwath, A. Torralba, and J. Glass, "Unsupervised learning of spoken language with visual context," in *NIPS*, 2016.

[35] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[37] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell, "Deep compositional captioning: Describing novel object categories without paired training data," in *CVPR*, 2016.

[38] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," in *ICASSP*, 2017.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] D. Hu, F. Nie, and X. Li, "Deep multimodal clustering for unsupervised audiovisual learning," in *CVPR*, 2019.

[41] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[43] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *ICRA*, 2016.

[44] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *CVPR*, 2016, pp. 4565–4574.

[45] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015, pp. 3128–3137.

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

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[48] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *ICML*, 2014, pp. 595–603.

[49] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *T-PAMI*, vol. 38, no. 1, pp. 14–29, 2015.

[50] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *NIPS*, 2018.

[51] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *CVPR*, 2017.

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

[53] Y. Li, "A deep spatiotemporal perspective for understanding crowd behavior," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3289–3297, 2018.

[54] ——, "Which way are you going? imitative decision learning for path forecasting in dynamic scenes," in *CVPR*, 2019, pp. 294–303.

[55] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," in *CVPR*, 2019, pp. 5725–5734.

[56] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *ECCV*, 2018.

[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[58] Y.-B. Lin, Y.-J. Li, and Y.-C. F. Wang, "Dual-modality seq2seq network for audio-visual event localization," in *ICASSP*, 2019.

[59] D. Liu, T. Jiang, and Y. Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in *CVPR*, 2019.

[60] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–a new baseline," in *CVPR*, 2018, pp. 6536–6545.

[61] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *CVPR*, 2019.

[62] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural baby talk," in *CVPR*, 2018, pp. 7219–7228.

[63] P. Luc, C. Couprie, Y. Lecun, and J. Verbeek, "Predicting future instance segmentation by forecasting convolutional features," in *ECCV*, 2018, pp. 584–599.

[64] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *CVPR*, 2016.

[65] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," *ICLR*, 2015.

[66] A. Miech, I. Laptev, J. Sivic, H. Wang, L. Torresani, and D. Tran, "Leveraging the present to anticipate the future in videos," in *CVPR-W*, 2019.

[67] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[68] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III, "Midge: Generating image descriptions from computer vision detections," in *EACL*, 2012, pp. 747–756.

[69] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in *CVPR*, 2018.

[70] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[71] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *ECCV*, 2018.

[72] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, "Visually indicated sounds," in *CVPR*, 2016.

[73] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning," in *ECCV*, 2016.

[74] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[75] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification," in *ECCV*, 2018.

[76] A. Piergiovanni, A. Wu, and M. S. Ryoo, "Learning real-world robot policies by dreaming," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7680–7687.

[77] T. Rahman, B. Xu, and L. Sigal, "Watch, listen and tell: Multi-modal weakly supervised dense event captioning," in *ICCV*, 2019.

[78] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *ICLR*, 2016.

[79] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[80] C. Rodriguez, B. Fernando, and H. Li, "Action anticipation by predicting future dynamic images," in *ECCV-W*, 2018.

[81] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," *NIPS-W*, 2016.

[82] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 1997.

[83] A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. So Kweon, "Learning to localize sound source in visual scenes," in *CVPR*, 2018.

[84] Y. Shi, B. Fernando, and R. Hartley, "Action anticipation with rbf kernelized feature mapping rnn," in *ECCV*, 2018, pp. 301–317.

[85] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *CVPR*, 2017.

[86] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos," in *ECCV*, 2018.

[87] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *CVPR*, 2016.

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

[89] ——, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2014.

[90] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *ICCV*, 2017.

[91] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI*, 2017.

[92] Y. Tian, C. Guan, J. Goodman, M. Moore, and C. Xu, "An attempt towards interpretable audio-visual video captioning," *arXiv preprint arXiv:1812.02872*, 2018.

[93] Y. Tian, C. Guan, G. Justin, M. Moore, and C. Xu, "Audio-visual interpretable and controllable video captioning," in *CVPR-W*, 2019.

[94] Y. Tian, D. Li, and C. Xu, "Unified multisensory perception: Weakly-supervised audio-visual video parsing," in *ECCV*, 2020.

[95] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu, "Audio-visual event localization in unconstrained videos," in *ECCV*, 2018.

[96] ——, "Audio-visual event localization in the wild," in *CVPR-W*, 2019.

[97] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[98] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *CVPR*, 2018.

[99] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[100] S. Venugopalan, L. Anne Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko, "Captioning images with diverse objects," in *CVPR*, 2017.

[101] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *ICCV*, 2015, pp. 4534–4542.

[102] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *NIPS*, 2016, pp. 3630–3638.

[103] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015, pp. 3156–3164.

[104] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," in *CVPR*, 2016.

[105] T. Vu, M. Liu, D. Phung, and G. Haffari, "Learning how to active learn by dreaming," in *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, 2019, pp. 4091–4101.

[106] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in *CVPR*, 2017.

[107] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.

[108] X. Wang, Y.-F. Wang, and W. Y. Wang, "Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning," in *NAACL*, 2018.

[109] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP*, 2019.

[110] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3d lstm: A model for video prediction and beyond," in *ICLR*, 2018.

[111] Q. Wu, C. Shen, P. Wang, A. Dick, and A. van den Hengel, "Image captioning and visual question answering based on attributes and external knowledge,"

*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1367–1381, 2017.

[112] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, "Visual question answering: A survey of methods and datasets," *Computer Vision and Image Understanding*, vol. 163, pp. 21–40, 2017.

[113] Y. Wu, L. Jiang, and Y. Yang, "Revisiting embodiedqa: A simple baseline and beyond," *IEEE Transactions on Image Processing*, vol. 29, pp. 3984–3992, 2020.

[114] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Bian, and Y. Yang, "Progressive learning for person re-identification with one example," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2872–2881, 2019.

[115] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning," in *CVPR*, 2018, pp. 5177–5186.

[116] Y. Wu and Y. Yang, "Exploring heterogeneous clues for weakly-supervised audio-visual video parsing," in *CVPR*, 2021.

[117] Y. Wu, L. Zhu, L. Jiang, and Y. Yang, "Decoupled novel object captioner," in *ACM MM*, 2018.

[118] Y. Wu, L. Zhu, X. Wang, Y. Yang, and F. Wu, "Learning to anticipate egocentric actions by imagination," *IEEE Transactions on Image Processing*, 2021.

[119] Y. Wu, L. Zhu, Y. Yan, and Y. Yang, "Dual attention matching for audio-visual event localization," in *ICCV*, 2019.

[120] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *CVPR*, 2018.

[121] 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.

[122] B. F. Yan Bin Ng, "Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting," *IEEE Transactions on Image Processing*, 2020.

[123] T. Yao, Y. Pan, Y. Li, and T. Mei, "Incorporating copying mechanism in image captioning for learning novel objects," in *CVPR*, 2017, pp. 5263–5271.

[124] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019.

[125] H. Zhao, C. Gan, W.-C. Ma, and A. Torralba, "The sound of motions," in *ICCV*, 2019.

[126] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, "The sound of pixels," in *ECCV*, 2018.

[127] P. Zhao, L. Xie, C. Ju, Y. Zhang, Y. Wang, and Q. Tian, "Bottom-up temporal action localization with mutual regularization," in *ECCV*, 2020.

[128] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *ICCV*, 2017.

[129] H. Zhou, X. Xu, D. Lin, X. Wang, and Z. Liu, "Sep-stereo: Visually guided stereophonic audio generation by associating source separation," in *ECCV*, 2020.

[130] L. Zhu, Z. Xu, Y. Yang, and A. G. Hauptmann, "Uncovering the temporal context for video question answering," *International Journal of Computer*

*Vision*, vol. 124, no. 3, pp. 409–421, Sep 2017.

[131] Y. Zhu, Y. Wu, H. Latapie, Y. Yang, and Y. Yan, "Learning audio-visual correlations from variational cross-modal generation," in *ICASSP*, 2021.

[132] Y. Zhu, Y. Wu, Y. Yang, and Y. Yan, "Describing unseen videos via multi-modal cooperative dialog agents," in *European Conference on Computer Vision*. Springer, 2020, pp. 153–169.