



**A USDOT NATIONAL  
UNIVERSITY TRANSPORTATION CENTER**

**Carnegie Mellon University**

---

## **Robust Automatic Detection of Traffic Activity**

**Alexander Hauptmann (PI)** (<https://orcid.org/0000-0003-2123-0684>)  
**Lijun Yu** (<https://orcid.org/0000-0003-0645-1657>)  
**Wenhe Liu** (<https://orcid.org/0000-0003-4679-2958>)  
**Yijun Qian** (<https://orcid.org/0009-0000-9440-9744>)  
**Zhiqi Cheng** (<https://orcid.org/0000-0002-1720-2085>)  
**Liangke Gui** (<https://orcid.org/0009-0009-0338-8948>)

**FINAL RESEARCH REPORT - June 30, 2023**

Contract # 69A3551747111

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. This report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. The U.S. Government assumes no liability for the contents or use thereof.

# CONTENTS

<b>Chapter 1 : Overview</b>	<b>1</b>
<b>Chapter 2 : <i>Argus++</i>: Robust Real-time Activity Detection for Unconstrained Video Streams with Overlapping Cube Proposals</b>	<b>3</b>
2.1 Introduction . . . . .	3
2.2 Related Work . . . . .	5
2.3 Method . . . . .	5
2.3.1 Activity Detection Task . . . . .	5
2.3.2 Argus++ System . . . . .	6
2.3.3 Proposal Generation . . . . .	6
2.3.4 Proposal Filtering . . . . .	7
2.3.5 Activity Recognition . . . . .	9
2.3.6 Activity Deduplication . . . . .	10
2.4 Experiments . . . . .	11
2.4.1 Implementation Details . . . . .	12
2.4.2 Evaluation Protocols . . . . .	12
2.4.3 ActEV Sequestered-Data Evaluation . . . . .	13
2.4.4 ActEV Self-Reported Evaluation . . . . .	13
2.4.5 ROAD Challenge . . . . .	13
2.4.6 Ablation Study . . . . .	14
2.5 Conclusion & Future work . . . . .	15
<b>Chapter 3 : <i>DAMO-StreamNet &amp; LongShortNet</i>: Pioneering Techniques for Optimizing Streaming Perception in Autonomous Driving</b>	<b>17</b>
3.1 Introduction . . . . .	17
3.2 Related Work . . . . .	19
3.3 DAMO-StreamNet . . . . .	19
3.3.1 Network Architecture . . . . .	20
3.3.2 Asymmetric Knowledge Distillation . . . . .	22
3.3.3 K-step Streaming Metric . . . . .	23
3.4 LongShortNet . . . . .	23
3.4.1 Long Short Fusion Module . . . . .	24
3.5 Experiments . . . . .	26
3.5.1 Dataset and Metric . . . . .	26

3.5.2	Implementation Details . . . . .	27
3.5.3	Comparison with State-of-the-art Methods . . . . .	27
3.5.4	Ablation Study . . . . .	28
3.6	Conclusion & Future work . . . . .	30
<b>Chapter 4 : Vehicle Tracking using Natural Language Descriptions</b>		<b>31</b>
4.1	Introduction . . . . .	31
4.2	Related Work . . . . .	32
4.2.1	Connecting Language and Vision . . . . .	32
4.2.2	Symmetric Network . . . . .	33
4.2.3	Multi-granularity Retrieval System . . . . .	34
4.3	Proposed Approach . . . . .	35
4.3.1	Data Augmentation . . . . .	36
4.3.2	Prompt Tuning . . . . .	36
4.3.3	Modified Symmetric Network . . . . .	38
4.3.4	SWIN Transformers . . . . .	40
4.4	Experiment Setup . . . . .	40
4.4.1	Dataset . . . . .	40
4.4.2	Evaluation . . . . .	40
4.4.3	Implementation Details . . . . .	41
4.5	Results . . . . .	41
4.6	Conclusion . . . . .	43
<b>Chapter 5 : Training Vision-Language Transformers from Captions</b>		<b>44</b>
5.1	Introduction . . . . .	44
5.2	Background . . . . .	44
5.3	Methodology . . . . .	45
5.3.1	Finetuning . . . . .	45
5.3.2	Inference . . . . .	47
5.4	Experiments . . . . .	49
5.5	Visualizations . . . . .	52
5.6	Conclusion . . . . .	52

## REFERENCES

# CHAPTER 1

## OVERVIEW

Nowadays, activity detection has drawn fast-growing attention in both industry and research fields. Activity detection in extended videos [15, 74] is widely applied for public safety in indoor and outdoor scenarios. Activity detection on streaming videos captured by in-vehicle cameras is applied for vision-based autonomous driving. The development of these applications brings several challenges. First, most of these systems take *unconstrained* videos as input, which is recorded in large field-of-views where multi-object and multi-activity occur simultaneously and continuously over time. Second, the unconstrained videos in the real world are in multiple scenarios and under multiple conditions, e.g. in dynamically changed road environments from day to night in autonomous driving [85]. Third, efficient algorithms are demanded for real-time processing and responding to streaming video.

Meanwhile, the advent of autonomous vehicles has driven the need for high-performance traffic environment perception systems. In this context, streaming perception, which involves detecting and tracking objects in a video stream simultaneously, is a fundamental technique that significantly impacts autonomous driving decision-making. Notably, the fast-changing scale of traffic objects due to vehicle motion can lead to conflicts in the receptive field when detecting both large and small objects. Moreover, real-time perception is an ill-posed problem that heavily depends on motion consistency context and historical data. Consequently, two major challenges in real-time perception are: (1) adaptively handling rapidly changing object scales, and (2) accurately and efficiently learning long-term motion consistency.

Therefore we developed two systems for enhancing traffic safety. The first system focuses on road activity detection, which identifies the activities of vehicles. We discuss the first system, Argus++, in Chapter 2. Further, we integrated the models in our video analysis framework *Argus++* to enable the real-time processing of traffic footage, including vehicle tracking. We introduce it in Chapter 4. Immediate notification could be provided on traffic density and speed estimation, and traffic incident detection. The second system focuses on streaming perception, which enhances the safety of autonomous driving. For this system, we introduce the models and algorithms in Chapter 3. Further, in Chapter 5, we discuss how to train a powerful model for our systems, especially training vision-language transformers from captions. We showed that this model could enhance the vehicle detection task.

The broader impacts of our real-time traffic video analysis system extend beyond its technical capabilities and immediate applications. They encompass the societal, economic, and environmental implications that the system may have. Here are some aspects to consider when discussing the broader impacts:

1. **Enhancing safety and security:** The action recognition system can contribute to public safety and security by detecting and analyzing suspicious or illegal activities in real-time. It can aid law enforcement agencies in monitoring public spaces, identifying potential threats or problems, and responding promptly to ensure public safety. This approach may also extend to other fields that are currently under-performing due to a lack of concerted effort of this type.
2. **Improving transportation and traffic management:** By accurately analyzing and predicting the actions and behaviors of vehicles, pedestrians, and cyclists, the system can help optimize traffic flow, reduce congestion, and enhance overall transportation efficiency. This can lead to improved travel times, reduced fuel consumption, and minimized environmental impact.
3. **Supporting urban planning and infrastructure development:** The insights provided by the system's analysis of traffic patterns, movement trends, and behavior can inform urban planners and policymakers in making informed decisions about transportation infrastructure development, traffic management strategies, and the allocation of resources.
4. **Enabling intelligent transportation systems:** Integration of the online action recognition system with existing transportation infrastructure and intelligent transportation systems can enable advanced functionalities such as adaptive traffic signal control, smart parking management, and dynamic routing. This can lead to improved traffic flow, reduced emissions, and enhanced overall transportation efficiency.
5. **Supporting emergency response:** In emergency situations or natural disasters, the system's ability to quickly detect abnormal or unusual actions can aid emergency response teams in identifying affected areas, assessing the impact, and coordinating rescue efforts. It can also assist in the evacuation of people and the allocation of resources.

# CHAPTER 2

## ARGUS++: ROBUST REAL-TIME ACTIVITY DETECTION FOR UNCONSTRAINED VIDEO STREAMS WITH OVERLAPPING CUBE PROPOSALS

### 2.1 INTRODUCTION

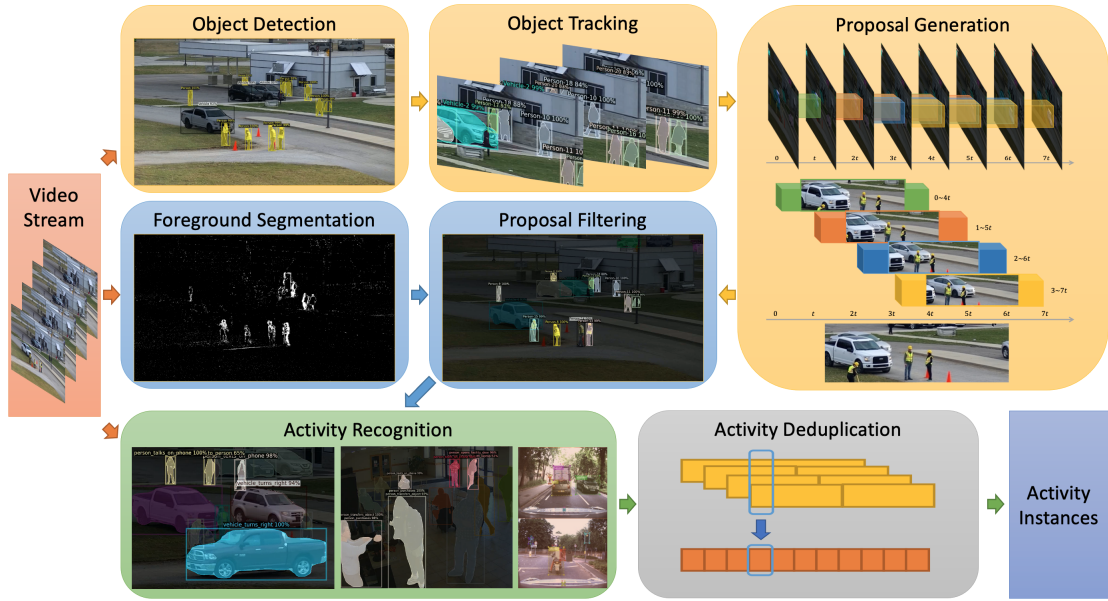


Figure 2.1: Architecture of *Argus++*. A video stream is processed frame-by-frame through object detection and tracking to generate overlapping cube proposals. With frame-level foreground segmentation, stable proposals are filtered out. Activity recognition models determine the classification scores for each proposal. These over-sampled cubes are deduplicated to produce the final activity instances.

Nowadays, activity detection has drawn a fast-growing attention in both industry and research fields. Activity detection in extended videos [15, 74] is widely applied for public safety in indoor and outdoor scenarios. Activity detection on streaming videos captured by in-vehicle cameras is applied for vision-based autonomous driving. The development of these applications brings several challenges. First, most of these systems take *unconstrained* videos as input, which are recorded in large field-of-views where multi-object and multi-activity occur simultaneously and continuously over time. Second, the unconstrained videos in real world are in multiple scenarios and under multiple conditions, e.g. in dynamically changed road environments from day to night in autonomous driving [85]. Third, efficient algorithms are demanded for real-time processing and responding of streaming video.

Conventional activity detection works [93, 24, 99, 46, 30] have achieved impressive performance. However, they are not suitable for real world unconstrained video understanding. Most of these works are applied under certain constraints, e.g., only for processing trimmed and/or object-centered video clips. Meanwhile, they usually are specified for certain scenarios, such as person activity, etc. Therefore, such algorithms would fail when being transferred to unconstrained videos on both efficiency and effectiveness.

Previous works [83, 112, 62] on unconstrained video analysis proposed to generate and analyze tube/tubelet proposals, which are trajectories extracted from object detection and tracking results. Tube proposal has several drawbacks. First, tube proposals failed to capture the trace of moving objects when cropping the proposals from the original videos. Therefore, learning the activities highly relied on trace would be difficult, e.g. 'vehicle turning right'. Second, the tube proposals still cannot stay away from temporal activity localization to determine the existence of the activities. Besides, most of the previous works [83] utilize non-overlapping proposals, which straightforwardly cuts the tube proposals by fixed length of temporal windows. Inevitably, such methods destroy the completeness of activities. Therefore, it would result in significant degrade of performance. Third, the objects in the tube proposal will suffer from the bounding box shift and distortion across frames, which could result in a high false alarm rate on activity detection.

To overcome the aforementioned challenges, we propose *Argus++*, an efficient robust spatio-temporal activity detection system for extended and road video activity detection. The proposed system contains four-stages: Proposal Generation, Proposal Filtering, Activity Recognition and Activity Deduplication. The major difference between *Argus++* and the former works, such as [62], is the concept of *cube* proposals. Rather than simply adapted tube proposals, i.e. cropped trajectories of detected and tracked objects, we propose to merge and crop the area of detected objects across the frames.

We summarize the contributions of our work as follows:

1. We propose *Argus++*, a real-time activity detection system for unconstrained video streams, which is robust across different scenarios.
2. We introduce overlapping spatio-temporal cubes as the core concept of activity proposals to ensure coverage and completeness of activity detection through over-sampling.
3. The proposed system has achieved outstanding performance in a large series of activity detection benchmarks, including CVPR ActivityNet ActEV 2021, NIST ActEV SDL UF/KF, TRECVID ActEV 2020/2021, and ICCV ROAD 2021.

## 2.2 RELATED WORK

**Object Detection and Tracking** Object detection and tracking are fundamental computer vision tasks that aims to detect and track objects from images or videos. Image-based object detection algorithms, such as Faster R-CNN [81] and R-FCN [16], have demonstrated convincing performance but are often expensive to apply on every frame. Video-based object detection algorithms [117, 79] use optical flow guided feature aggregation to leverage motion information and reduce computation. With the deep features extracted from the backbone convolutional network, multi-object tracking algorithms [103, 102] associates objects across frames based on feature similarity and location proximity.

**Activity Detection** In recent years, there emerged some systems designed for spatio-temporal activity detection on unconstrained videos [83, 112, 62, 8, 110, 114]. Generally, these systems first generates activity proposals and then feeds them to classification models. Since there have been a variety of video classification networks [93, 54, 24], the major focus is on the paradigm of proposals and the generation algorithm. In [62, 8], a detection and tracking framework is employed to extract whole object tracklets as tubelets, where temporal localization is required. In [83], an encoder-decoder network is used to generate localization masks on fixed-length clips for tubelet proposal extraction, which has varied spatial locations in different frames.

## 2.3 METHOD

### 2.3.1 Activity Detection Task

In this paper, we tackle the activity detection task in unconstrained videos which are untrimmed and with large field-of-views. Given an untrimmed video stream  $\mathcal{V}$ , the system  $\mathcal{S}$  should identify a set of activity instances  $\mathcal{S}(\mathcal{V}) = \{A_i\}$ . Each activity instance is defined by a three-tuple  $A_i = (T_i, L_i, C_i)$ , referring to an activity of type  $C_i$  occurs at temporal window  $T_i$  with spatial location  $L_i$ .  $L_i$  contains the precise location of  $A_i$  in each frame, forming a tube in the timeline. As such, activity detection can often be decomposed into three aspects, i.e., temporal localization ( $T_i$ ), spatial localization ( $L_i$ ), and action classification ( $C_i$ ).

Each of the three aspects poses unique challenges to the video understanding system. Due to its multi-dimensional nature, it remains hard to define and build a useful activity detection system under the strict setting. Therefore, we also evaluates with some loosened requirements.



**Strict Setting** All activity types are defined as atomic activities with clear temporal boundaries and spatial extents. The evaluation metric performs bipartite matching between predictions and ground truths.

**Loosened Setting** Activity types are either atomic activities within a temporal window (e.g. standing up) or continuous repetitive activities that can be cut into multiple identifiable windows (e.g. walking). The evaluation metric allows multiple non-overlapping predictions to be matched with one ground truth.

### 2.3.2 Argus++ System

The architecture of the proposed *Argus++* system is shown in Figure 2.1. To tackle the task of activity detection, we adopt an intermediate concept of *spatio-temporal cube proposal* with a much simpler definition than an activity instance:

$$p_i = (x_0^i, x_1^i, y_0^i, y_1^i, t_0^i, t_1^i) \quad (2.1)$$

This six-tuple design relieves the localization precision and caters modern action classification models which works on fixed-length clips with fixed spatial window.

For an input video stream, the system first generates candidate proposals with frame-wise information such as detected objects, which will be covered in Section 2.3.3. These proposals are filtered with a background subtraction model as detailed in Section 2.3.4. Then, action recognition models described in Section 2.3.5 are applied on the proposals to predict per-class confidence scores. Finally, Section 2.3.6 introduces the post-processing stage to merge and filter the proposals with scores and generate final activity instances.

### 2.3.3 Proposal Generation

Starting this section, we introduce each of the components of *Argus++*. The system begins by generating a set of cube proposals. They are generated based on information from frame-level object detection with multiple object tracking methods. Cubes are sampled densely in the timeline with refined spatial locations.

**Detection and Tracking** To conduct activity recognition, we first locate the candidate objects (in most cases, person and vehicle) in the video. For each selected frame  $F_i$ , we apply an object detection model to get objects  $O_i = \{o_{i,j} \mid j = 1, \dots, n_i\}$  with object types  $c_{i,j}$  and bounding boxes  $(x_0, x_1, y_0, y_1)_{i,j}$ . Objects are detected in a stride of every  $S_{det}$  frames. A multiple object tracking algorithm is applied on the detected objects to assign track ids to each of them as  $tr_{i,j}$ .

**Proposal Sampling** To sample proposals on untrimmed videos without breaking the completeness of any activity instances, we propose a dense overlapping proposals sampling algorithm. As illustrated in Figure 2.2, this method ensures coverage of activities occurring at any time, with no hard boundaries. Two parameters, duration  $D_{prop}$  and stride  $S_{prop}$ , controls the sampling process. Each proposal contains a temporal window of  $D_{prop}$  frames. New proposals are generated every  $S_{prop} \leq D_{prop}$  frames, possibly with overlaps. Generally, non-overlapping proposal system can be treated as a degraded case when  $S_{prop} = D_{prop}$ .

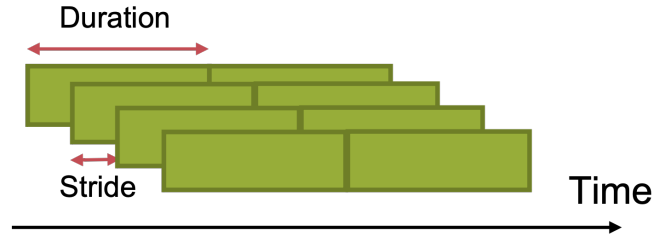


Figure 2.2: Dense Overlapping Proposals

**Proposal Refinement** To generate proposals in a temporal window from  $t_0$  to  $t_1 = t_0 + D_{prop}$ , we select seed track ids  $Tr_{t_c}$  from the central frame  $t_c = \lfloor \frac{t_0+t_1}{2} \rfloor$ . Their bounding boxes are enlarged as the union across the temporal window

$$(x_0, x_1, y_0, y_1)_k = \bigcup (\{(x_0, x_1, y_0, y_1)_{i,j} \mid t_0 \leq i \leq t_1, tr_{i,j} = tr_{t_c,k}\}) \quad (2.2)$$

$$k = 1, \dots, n_{t_c}$$

This algorithm is robust through identity switch in the tracking algorithm as it uses the stable seeds from the central frame. It also ensures the coverage of moving objects by enlarging the bounding box when it's successfully tracked. This design is helpful for efficiency optimization by allowing a large detection stride  $S_{det}$ . When later applied for activity recognition, the bounding box can be further enlarged for a fixed rate  $R_{enl}$  to include spatial context and compensate for missed tracks.

### 2.3.4 Proposal Filtering

For now, the proposal generation pipeline applies a frame-wise object detection with slight aid of tracking information. The motion information of video is not yet explored. To produce high quality proposals, we apply a proposal filtering algorithm to eliminate the proposals that are unlikely to contain activities.

**Foreground Segmentation** For each proposal, a foreground segmentation algorithm is implemented to generate a binary mask for every  $S_{bg}$  frames for each video clip. We

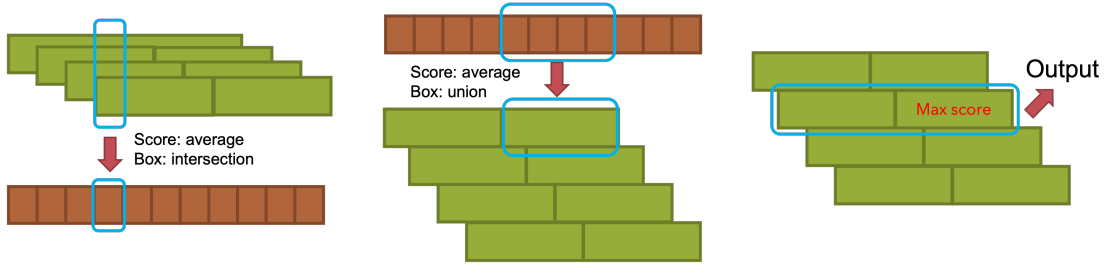


Figure 2.3: Deduplication Algorithm for Overlapping Proposals

average the value of pixel masks in its cube to get its foreground score  $f_i$ . For proposals generated by object type  $c$ , those proposals with  $f_i \leq F_c$  will be filtered out. The threshold  $F_c$  is determined by allowing up to  $P_{pos}$  true proposals to be filtered out.

**Label Assignment** To determine the above threshold and to train the activity recognition module, we need to assign labels for each generated proposal according to the ground truth activity instances. We first convert the annotation of activity instances into the cube format, denoted as ground truth cubes, by performing dense sampling of duration  $D_{prop}$  and stride  $S_{prop}$  within each instance. For each proposal, we estimate the spatial intersection-over-union (IoU) between it and ground truth cubes in the same temporal window. Then we follow Faster R-CNN [81] in the assignment process:

- For each ground truth cube, assign it to the proposal with the highest score above  $S_{low}$ .
- For each proposal, assign it with each ground truth cube with score above  $S_{high}$ .
- For each proposal, assign it as negative if all scores are below  $S_{low}$ .

$S_{high}$  and  $S_{low}$  are the high and low thresholds. Through this algorithm, each proposal may be assigned one or more positive labels, a negative label, or nothing. Those assigned nothing are redundant detections which will not be used in classifier training.

**Proposal Evaluation** To measure the quality of proposals before and after the filtering, we need a method for proposal evaluation. This can be achieved by assuming a perfect classifier in the activity recognition part, so the final metrics reflects the upper bound performance with current proposals. To do this, we simply use the assigned labels as the classification outputs and pass through the deduplication algorithm covered later. To further measure other properties of the generated proposals, we can only pass through a subset of them, such as only those with spatial IoU against ground truth above 0.5.

<sup>1</sup><http://activity-net.org/challenges/2021/challenge.html>

<sup>2</sup>[https://actev.nist.gov/sdl#tab\\_leaderboard](https://actev.nist.gov/sdl#tab_leaderboard)

Table 2.1: CVPR 2021 ActivityNet Challenge<sup>1</sup> ActEV SDL Unknown Facility Evaluation

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean $P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
<b>Argus++ (Ours)</b>	<b>0.3535</b>	<b>0.5747</b>	0.576
UMD_JHU	<u>0.4232</u>	0.6250	0.345
IBM-Purdue	0.4238	0.6286	0.530
UCF	0.4487	<u>0.5858</u>	0.615
Visym Labs	0.4906	<u>0.6775</u>	0.770
MINDS_JHU	0.6343	0.7791	0.898

Table 2.2: NIST ActEV’21 SDL<sup>2</sup>Known Facility Evaluation

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean $P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
<b>Argus++ (Ours)</b>	<b>0.1635</b>	<b>0.3424</b>	0.413
UCF	<u>0.2325</u>	<u>0.3793</u>	0.751
UMD	0.2628	0.4544	0.380
IBM-Purdue	0.2817	0.4942	0.631
Visym Labs	0.2835	0.4620	0.721
UMD-Columbia	0.3055	0.4716	0.516
UMCMU	0.3236	0.5297	0.464
Purdue	0.3327	0.5853	0.131
MINDS_JHU	0.4834	0.6649	0.967
BUPT-MCPRL	0.7985	0.9281	0.123

### 2.3.5 Activity Recognition

In this section, we will elaborately introduce our action recognition modules. Given the input proposal of an activity instance  $p_i$ , our action recognition model  $\mathbb{V}$  will give out the confidence vector  $c_i$ :

$$\mathbb{V}(p_i) = c_i = \{c_i^1, c_i^2, \dots, c_i^n\} \quad (2.3)$$

Where  $n$  represents the number of target actions, and  $c_i \in \mathbb{R}^n$ . Limited by GPU memory size and temporal length settings of pretrained weights, we need to select  $t$  frames out of  $t_1^i - t_0^i$  samples from the activity instance. To do this, we strictly followed the sparse-sampling strategy mentioned in [99] for both training and inference stage. To be specific, the video is evenly separated into  $t$  segments. From each segment, 1 frame will be randomly selected to generate the sampled clip.

To transform the action recognition modules from previous multi-class task to the realm of multi-label recognition, we modified the loss function for optimization. Instead of traditional cross entropy loss (XE), we implemented a weighted binary cross entropy loss (wBCE). In which, two weight parameters are adopted, the activity-wise weight  $W_a = \{w_a^1, w_a^2, \dots, w_a^n\}$  and the positive-negative weight  $W_p = \{w_p^1, w_p^2, \dots, w_p^n\}$ .  $W_a$  balances the training samples of different activities and  $W_p$  balances the positive and negative samples of a specific activity. With the aligned label sequence of  $i^{th}$  instance

represented as  $Y_i = \{y_i^1, y_i^2, \dots, y_i^n\} \in \mathbb{R}^n$ . The calculation of  $w_a^c$  is derived as:

$$\hat{w}_a^c = \frac{1}{\sum_{i \in [I]} y_i^c} \quad (2.4)$$

$$w_a^c = n \times \frac{\hat{w}_a^c}{\sum_{c \in [n]} \hat{w}_a^c} \quad (2.5)$$

And the derivation of  $w_p^c$  is:

$$w_p^c = \frac{\sum_{i \in [I]} \mathbf{1}_{y_i^c=0}}{\sum_{i \in [I]} y_i^c} \quad (2.6)$$

In which,  $[I]$  represents all input instances, and  $[n]$  represent all target activities. Compared with vanilla BCE loss, we found wBCE loss can significantly improve the final performance on internal validation set.

Furthermore, we tried multiple action recognition modules and made late fusion action-wisely according to the results on the validation set. We found each classifier does show superiority on certain actions. Through the feedback from the online leaderboard, such fusion strategy can improve the final performance with noticeable margins.

Table 2.3: NIST ActEV’21 SDL Unknown Facility Evaluation

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean $P_{miss}@0.02T_{fa} \downarrow$	Relative Processing Time
<b>Argus++ (Ours)</b>	<b>0.3330</b>	<u>0.5438</u>	0.776
UCF	<u>0.3518</u>	<b>0.5372</b>	0.684
IBM-Purdue	0.3533	0.5531	0.575
Visym Labs	0.3762	0.5559	1.027
UMD	0.3898	0.5938	0.515
UMD-Columbia	0.4002	0.5975	0.520
UMCMU	0.4922	0.6861	0.614
Purdue	0.4942	0.7294	0.239
MINDS_JHU	0.6343	0.7791	0.898

### 2.3.6 Activity Deduplication

**Overlapping Instances** As the system generates overlapping proposals, it could have duplicate predictions for some of the proposals. This would result in a large amount of false alarms unless we deduplicate them. Figure 2.3 is a diagram for our deduplication algorithm which applies to each activity type with all proposals:

1. Split the overlapping cubes of duration  $D_{prop}$  and stride  $S_{prop}$  into non-overlapping cubes of duration  $S_{prop}$ . An output cube relies on all original cubes in the temporal window, with an averaged score and an intersected bounding box.
2. Merge the non-overlapping cubes of duration  $S_{prop}$  back into  $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$  groups of

non-overlapping cubes of duration  $D_{prop}$ . An output cube is merged from  $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$  cubes with an averaged score and the union of bounding boxes.

3. Select the group where the maximum score resides.

The deduplication algorithm performs an interpolation upon the overlapping cubes. Each group in step 3 contains information from every classification results, maximizing the information utilization.

**Adjacent Instances** The above deduplication process only transforms overlapping instances to non-overlapping instances with the same duration. This would be sufficient under the *Loosened Setting*, where multiple predictions are allowed for each activity. No threshold would be needed to truncate low-confidence predictions as this happens automatically during the ground-truth matching process.

However, for the *Strict Setting*, we need to further merge adjacent cubes into integrate instances. Currently we adopt a simple yet effective algorithm, by simply merging adjacent cubes where all of them have confidence score above  $S_{merg}$ . The merged instance needs to be longer than  $L_{merg}$  to be kept in the final output.

## 2.4 EXPERIMENTS

Table 2.4: NIST TRECVID 2021 ActEV Evaluation [1, 113]

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean $P_{miss}@0.15T_{fa} \downarrow$	Mean $wP_{miss}@0.15R_{fa} \downarrow$
<b>Argus++ (Ours)</b>	<b>0.39607</b>	<b>0.30622</b>	<b>0.81080</b>
BUPT	0.40853	0.32489	<b>0.79798</b>
UCF	0.43059	0.34080	0.86431
M4D	0.84658	0.79410	0.88521
TokyoTech_AIST	0.85159	0.81970	0.94897
Team UEC	0.96405	0.95035	0.95670

Table 2.5: NIST TRECVID 2020 ActEV Evaluation [2, 112]

System/Team	$nAUDC@0.2T_{fa} \downarrow$	Mean $P_{miss}@0.15T_{fa} \downarrow$	Mean $wP_{miss}@0.15R_{fa} \downarrow$
<b>Argus++ (Ours)</b>	<b>0.42307</b>	<b>0.33241</b>	<b>0.80965</b>
UCF	0.54830	0.50285	0.83621
BUPT-MCPRL	0.55515	0.48779	0.84519
TokyoTech_AIST	0.79753	0.75502	0.87889
CERTH-ITI	0.86576	0.84454	0.88237
Team UEC	0.95168	0.95329	0.98300
Kindai_Kobe	0.96267	0.95204	0.93905

### 2.4.1 Implementation Details

In *Argus++*, we apply Mask R-CNN [38] with a ResNet-101 [39] backbone from Detectron2 [104] pre-trained on the Microsoft COCO dataset [58] as the object detector, with  $S_{det} = 8$ . Only person, vehicle, and traffic light classes are selected. For the tracking algorithm, we apply the work in [102] and reuse the region-of-interest from the ResNet backbone as in [111, 77].

The proposals are generated with  $D_{prop} = 64$  and  $S_{prop} = 16$ . The labels are assigned with  $S_{high} = 0.5$  and  $S_{low} = 0$ . The proposal filter is set with a tolerance of  $P_{pos} = 0.05$ .

For activity classifiers, we adopted multiple state-of-the-art models including R(2+1)D [93], X3D [24], and Temporal Relocation Module (TRM) [78]. During training procedure, frames are cropped with jittering [99] and enlarged with  $R_{enl} = 0.13$ . For X3D and TRM, we trained modules with weights pre-trained on Kinetics [46]. For R(2+1)D modules, we trained modules with weights pre-trained on IG65M [30]. We fused confidence scores from these models according to their performance on the validation set.

### 2.4.2 Evaluation Protocols

To measure the performance, efficiency, and generalizability of *Argus++*, we evaluate it across a series of public benchmarks. *Argus++* is applied to NIST Activities in Extended Videos (ActEV) evaluations on MEVA [15] Unknown Facility, MEVA Known Facility, and VIRAT [74] settings for surveillance activity detection. With slight modifications, it is also tested in the ICCV 2021 ROAD challenge for the action detection task in autonomous driving.

In the NIST evaluations, the metrics [2] are designed in the *Loosened Setting*, where short-duration outputs are allowed and spatial alignment is ignored. The idea was that, after processed by the system, there will still be human reviewers to inspect the activity instances with the highest confidence scores for further usages. The performance is thus measured by the probability of miss detection ( $P_{miss}$ ) of activity instances within a time limit of all positive frames plus  $T_{fa}$  of negative frames, where  $T_{fa}$  is referred to as time-based false alarm rate. The major metric,  $nAUDC@0.2T_{fa}$ , is an integration of  $P_{miss}$  on  $T_{fa} \in [0, 0.2]$ .

In the ROAD challenge, the *Strict Setting* is adopted by using the mean average precision (mAP) at 3D intersection-over-union (IoU) evaluation metric. This metric does exact bipartite matching between predictions and ground truth instances, with challenging localization precision requirements.

For metrics in the following tables,  $\downarrow$  means lower is better and  $\uparrow$  means higher is

better. For each metric, the best value is bolded and the second best is underscored. For ongoing public evaluations, the result snapshot at 11/01/2021 is presented.

### 2.4.3 ActEV Sequestered-Data Evaluation

ActEV Sequestered Data Leaderboards (SDL) are platforms where a system is submitted to run on NIST’s evaluation servers. This submission format prevents access to the test data and measures the processing time with unified hardware platform<sup>3</sup>. For these evaluations, *Argus++* was trained on MEVA, a large-scale surveillance video dataset with activity annotations of 37 types. We used 1946 videos in its training release drop 11 as the training set and 257 videos in its KF1 release as validation set. The optimization target is reaching better performance within 1x real-time.

Table 2.1 shows the published results from CVPR 2021 ActivityNet Challenge ActEV SDL Unknown Facility evaluation, where *Argus++* demonstrated around 20% advantage in  $nAUDC@0.2T_{fa}$  over runner-up system. The test set of unknown facility is captured with a different setting from MEVA, which challenges the generalization of action detection models. Table 2.2 shows the ongoing NIST ActEV’21 SDL Known Facility leaderboard, where *Argus++* shows over 40% advantage in  $nAUDC@0.2T_{fa}$ . The test set of known facility shares a similar distribution with MEVA, where our system learns well and is getting nearer for real-world usages. Table 2.3 shows the ongoing NIST ActEV’21 SDL Unknown Facility leaderboard continued from ActivityNet, where *Argus++* still holds the leading position with over 5% advantage in  $nAUDC@0.2T_{fa}$ .

### 2.4.4 ActEV Self-Reported Evaluation

ActEV self-reported evaluations are where only results are submitted and test data is accessible. This currently includes the annual TRECVID ActEV evaluations on VIRAT. For TRECVID, we use the official splits of VIRAT for training and validation.

Table 2.4 and 2.5 shows the leaderboard of 2020 and 2021 NIST TRECVID ActEV Challenge. In 2020, our systems is 22.8% better in  $nAUDC@0.2T_{fa}$ , 33.8% better in Mean  $P_{miss}@0.15T_{fa}$ , and 3.5% better in Mean- $wP_{miss}@0.15R_{fa}$  than the runner-up. Although the other competitors improved significantly in 2021, our system still holds the first place with noticeable margins.

### 2.4.5 ROAD Challenge

Different from previous surveillance action detection benchmarks, the videos of ROAD Challenge[69] are gathered from the point of view of autonomous vehicles. It contains

---

<sup>3</sup>[https://actev.nist.gov/pub/Phase3\\_ActEV\\_2021\\_SDL\\_EvaluationPlan\\_20210803.pdf](https://actev.nist.gov/pub/Phase3_ActEV_2021_SDL_EvaluationPlan_20210803.pdf)



122K frames from 22 annotated videos, where each video is 8 minutes long on average. Totally 7K tubes of individual agents are included and each tube consists on average of approximately 80 bounding boxes linked over time.

Table 2.6 shows the performance of our system with other competitors. Our system ranks the first with 20% average mAP. Although the performance is still far from satisfying in this *Strict Setting*, it demonstrates the capability of *Argus++* in adapting to precise 3D localization and moving camera view points.

Table 2.6: ICCV 2021 ROAD Challenge Action Detection<sup>4</sup>

System/Team	Action@0.1 $\uparrow$	Action@0.2 $\uparrow$	Action@0.5 $\uparrow$	Average $\uparrow$
<b>Argus++ (Ours)</b>	<b>28.54</b>	<b>25.63</b>	6.98	<b>20.38</b>
THE IFY	<u>28.15</u>	<u>20.97</u>	6.58	<u>18.57</u>
YAAHO	26.81	20.40	<u>7.02</u>	18.07
hyj	26.52	20.32	<b>7.05</b>	17.97
3D RetinaNet [85]	25.70	19.40	6.47	17.19
LeeC	13.64	9.89	2.23	8.59

Table 2.7: Proposal Quality Metrics on VIRAT Validation Set

$nAUDC@0.2T_{fa}$ Threshold	IoU			Reference Coverage		
	Average	$\geq 0$	$\geq 0.5$	Average	$\geq 0.5$	$\geq 0.9$
Unfiltered Proposals	0.2358	0.0772	0.1518	0.1562	0.1125	0.4211
Filtered Proposals	0.2352	0.0772	0.1469	0.1563	0.1099	0.4280

#### 2.4.6 Ablation Study

**Coverage of Proposal Formats** We analyze the coverage of dense spatio-temporal proposals and determines the best hyper-parameters for the proposal format. By directly use ground truth cubes as proposals, we estimate the upper bound performance of both overlapping and non-overlapping proposal formats on VIRAT validation set. The results are shown in Table 2.8, where non-overlapping proposals shows at least 6.7% systematic errors while overlapping proposals with duration 64 and stride 16 only has 1.3%.

**Performance of Proposal Filtering** We examine the quality of the proposals with and without the filter, as shown in Table 2.9 and 2.7. With the proposal evaluation procedure introduced in Section 2.3.4, the proposals are further filtered by IoU with reference and coverage of reference at levels from 0, 0.1, to 0.9 to calculate partial results.

<sup>4</sup><https://eval.ai/web/challenges/challenge-page/1059/leaderboard/2748>

Table 2.8: Lower Bounds of  $nAUDC@0.2T_{fa}$  on VIRAT Validation Set with different proposal formats. *Italic values are non-overlapping proposals while the others are overlapping proposals.* Duration and stride are in the unit of frames.

Duration / Stride	16	32	64	96
32	0.0705	<i>0.1208</i>	-	-
64	<b>0.0127</b>	0.0621	<i>0.0673</i>	-
96	0.0275	0.0504	-	<i>0.0688</i>

Table 2.9: Statistics of Proposals on VIRAT Validation Set

Name	Unfiltered	Filtered
Number of Proposals	211271	62831
Positive rate	0.1704	<b>0.5204</b>
Rate of unique label	0.4558	0.4415
Rate of two labels	0.4127	0.4252
Rate of three labels	0.1017	0.1060

With the dense cube proposals, the best  $nAUDC@0.2T_{fa}$  we can achieve with a ideal classifier is 0.08, as indicated in the  $IoU \geq 0$  column. The  $IoU$  and reference coverage bounded scores are used to measure the spatial matching quality of proposals, as the  $nAUDC@0.2T_{fa}$  does not consider spatial alignments. We can see that even with a condition of  $IoU \geq 0.5$ , our proposal can achieve up to 0.15, which indicates the spatial preciseness. The proposal filter is also proved effective, which removed 70% of original proposals without dropping the recall level.

The effect of the proposal filter is also evaluate on the SDL, as shown in Table 2.10. It not only reduces processing time from 0.925 to 0.582, but also improves  $nAUDC@0.2T_{fa}$  due to reduced false alarms.

Table 2.10: Proposal Filter on NIST ActEV’21 SDL Unknown Facility Micro Set

Proposal Filter	$nAUDC@0.2T_{fa} \downarrow$	Processing Time
<b>Enabled</b>	<b>0.4822</b>	0.582
Disabled	0.5176	0.925

## 2.5 CONCLUSION & FUTURE WORK

In this work, we proposed *Argus++*, a robust real-time activity detection system for analyzing unconstrained video streams. We introduced *overlapping spatio-temporal cubes* as an intermediate concept of activity proposals to ensure coverage and completeness of activity detection through over-sampling. The proposed system is able to process unconstrained videos with robust performance across multiple scenarios and

real-time efficiency on consumer-level hardware. Extensive experiments on different surveillance and driving scenarios demonstrated its superior performance in a series of activity detection benchmarks, including CVPR ActivityNet ActEV 2021, NIST ActEV SDL UF/KF, TRECVID ActEV 2020/2021, and ICCV ROAD 2021.

Future works are suggested to focus on extending the current system to more applications, such as action detection in UAV captured videos, first-person human activity understanding, etc. The proposed system could also be extended to end-to-end frameworks for better performance.

## CHAPTER 3

# DAMO-STREAMNET & LONGSHORTNET: PIONEERING TECHNIQUES FOR OPTIMIZING STREAMING PERCEPTION IN AUTONOMOUS DRIVING

### 3.1 INTRODUCTION

The advent of autonomous vehicles has driven the need for high-performance traffic environment perception systems. In this context, streaming perception, which involves detecting and tracking objects in a video stream simultaneously, is a fundamental technique that significantly impacts autonomous driving decision-making. Notably, the fast-changing scale of traffic objects due to vehicle motion can lead to conflicts in the receptive field when detecting both large and small objects. Moreover, real-time perception is an ill-posed problem that heavily depends on motion consistency context and historical data. Consequently, two major challenges in real-time perception are: (1) adaptively handling rapidly changing object scales, and (2) accurately and efficiently learning long-term motion consistency.

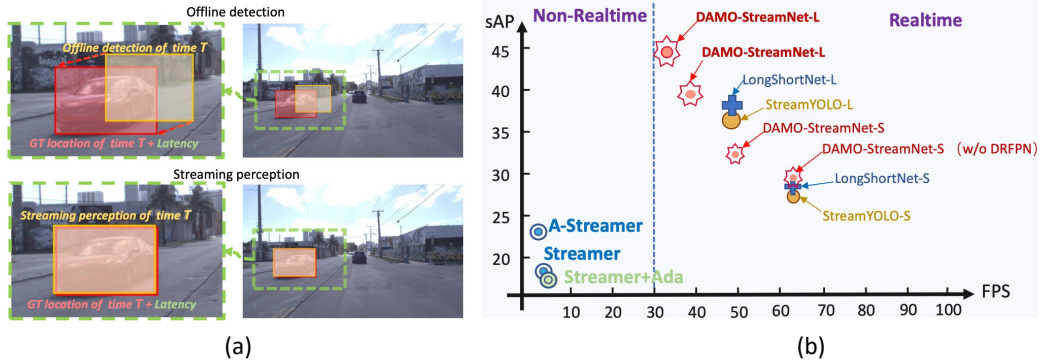


Figure 3.1: Comparison of offline detection (VOD) and streaming perception, where the latter is real-time and can respond promptly to motion changes (a), and performance comparisons of streaming perception task, showcasing the balance between accuracy and speed achieved by our proposed methods, DAMO-StreamNet [34] and LongShortNet [51], which sets a new state-of-the-art benchmark (b).

Despite previous research on temporal aggregation techniques [100, 9, 55, 88, 42] has primarily focused on offline settings and is unsuitable for online real-time perception. Furthermore, enhancing the base detector has not been thoroughly investigated in the context of real-time perception. To address these limitations, we propose DAMO-StreamNet [34] and LongShortNet [51], two practical real-time perception pipelines that improve the model in four key aspects:

1. *To enhance the base detector’s performance*, we propose an efficient feature aggregation scheme called Dynamic Receptive Field FPN. This scheme utilizes connections and deformable convolution networks to resolve receptive field conflicts and bolster feature alignment capacity. We also adopt the cutting-edge detection technique Re-parameterized to further enhance the network’s performance without adding extra inference costs. *These improvements lead to higher detection accuracy and faster inference times.*
2. *To capture long-term spatial-temporal correlations*, we design a dual-path structure temporal fusion module. *This module employs a two-stream architecture that separates spatial and temporal information, facilitating the accurate and efficient capture of long-term correlations.*
3. *To tackle the challenges of learning long-term motion consistency*, we propose an Asymmetric Knowledge Distillation (AK-Distillation) framework. This framework employs a teacher-student learning strategy in which student networks are supervised by transferring the generalized knowledge captured by large-scale teacher networks. *This method enforces the long-term motion consistency of the feature representations between the teacher-student pair, resulting in enhanced performance.*
4. *To fulfill the real-time forecasting requirement*, we update the support frame features with the current frame before the next prediction in the inference phase. Additionally, the support frame features are updated by the current frame to prepare for the next prediction in the inference phase to satisfy the real-time forecasting requirement. *This approach allows for the pipeline to handle real-time streaming perception and make predictions on time.*

*In summary, our methods offer a state-of-the-art solution for real-time perception in autonomous driving. DAMO-StreamNet [34] outperforms existing SOTA methods, achieving 37.8% (normal size (600, 960)) and 43.3% (large size (1200, 1920)) sAP without using any extra data. LongShortNet [51] achieves 37.1% (normal size (600, 960)) and 42.7% (large size (1200, 1920)) sAP without using any extra data, outperforming the existing state-of-the-art StreamYOLO [108] with almost the same time cost (20.23 ms vs. 20.12 ms). Our work not only establishes a new benchmark for real-time perception but also provides valuable insights for future research in this field. Moreover, DAMO-StreamNet<sup>1</sup> and LongShortNet<sup>2</sup> can be applied to various types of autonomous systems, such as drones and robots, enabling real-time and accurate environmental perception.*

<sup>1</sup>DAMO-StreamNet is at <https://github.com/zhiqic/DAMO-StreamNet>.

<sup>2</sup>LongShortNet is at <https://github.com/zhiqic/LongShortNet>.

## 3.2 RELATED WORK

**Image Object Detection.** In recent years, remarkable progress in deep learning-based object detection has been witnessed. Image object detection is fundamental to streaming perception. Therefore, we first review the state-of-the-art detectors [29, 97] and cutting-edge techniques from multiple aspects, including backbone design [98, 19, 20, 18, 96], effective feature aggregation [57, 31, 44, 91, 11, 94], and optimal label assignment [28, 47, 7]. Associated with the backbone network development, the feature aggregation solution, FPN [57] and PAFPN [61] are known as ‘necks’ in the general detection pipeline. Neural Architecture Search (NAS) is also applied to this topic, introducing NAS-FPN [31, 12, 41] for object detection. All the efforts aforementioned are mainly for bridging the representation gap between classification and object detection. Beyond this setting, GiraffeDet [44] adopts an extremely lightweight backbone but a heavy neck for feature learning.

**Video Object Detection.** A common schema to learn the temporal dynamics is feature aggregation which boosts per-frame feature representation by aggregating the features of nearby frames [100, 9, 55, 88, 50]. DeepFlow [119] and FGFA [118] utilize the optic flow from FlowNet [22] to model motion relations via different temporal feature aggregation. MANet [100] self-adaptively combines pixel-level and instance-level calibration according to the motion in a unified framework to calibrate the features at pixel-level with inaccurate flow estimation. Despite the gratifying success of these approaches, most of the pipelines for video object detection are overly sophisticated, requiring extra temporal modeling components, e.g., optical flow model [119], recurrent neural network [55, 36], feature alignment module [105, 80, 35], relation networks [27]. An effective and simple way for VOD is by adopting a temporal linking module such as Seq-NMS [33], Tubelet rescoring [45] and Seq-Bbox Matching [5, 50] as post-processing, which links the same object across the video to form tubelets and aggregating classification scores to achieve the state-of-the-art performance.

## 3.3 DAMO-STREAMNET

The overall framework is illustrated in Fig. 3.4. Initially, a video frame sequence passes through DAMO-StreamNet to extract spatiotemporal features and generate the final output feature. Subsequently, the Asymmetric Knowledge Distillation module (AK-Distillation) takes the output logit features of the teacher and student networks as inputs, transferring the semantics and spatial position of the future frame extracted by the teacher to the student network.

Given a video frame sequence  $\mathcal{S} = \{I_t, \dots, I_{t-N\delta t}\}$ , where  $N$  and  $\delta t$  represent the

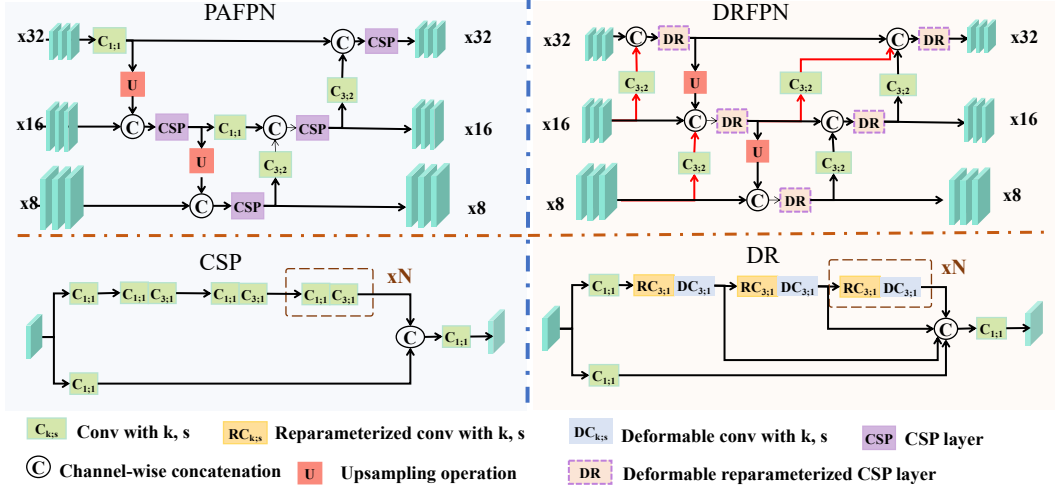


Figure 3.2: A comprehensive comparison between PAFPN and our proposed DRFPN, both constructed using the base block CSP and DR layer. The notation ‘‘Conv with  $k, s$ ’’ represents a convolution layer with kernel size ‘ $k$ ’ and stride ‘ $s$ ’.

number and step size of the frame sequence, respectively. DAMO-StreamNet can be defined as,

$$\mathcal{T} = \mathcal{F}(\mathcal{S}, W),$$

where  $W$  denotes the network weights, and  $\mathcal{T}$  represents the collection of final output feature maps.  $\mathcal{T}$  can be further decoded using  $Decode(\mathcal{T})$  to obtain the result  $\mathcal{R}$ , which includes the score, category, and location of the objects.

In the training phase, the student network can be represented as,

$$\mathcal{T}_{stu} = \mathcal{F}_{stu}(\mathcal{S}, W_{stu}).$$

Besides the student network, the teacher network takes the  $t + 1$  frame as input to generate the future result, represented by,

$$\mathcal{T}_{tea} = \mathcal{F}_{tea}(I_{t+1}, W_{tea}),$$

where  $W_{stu}$  and  $W_{tea}$  denote the weights of the student and teacher networks, respectively. Then, AK-Distillation leverages  $\mathcal{T}_{stu}$  and  $\mathcal{T}_{tea}$  as inputs to perform knowledge distillation  $AKDM(\mathcal{T}_{stu}, \mathcal{T}_{tea})$ . More details are elaborated in the following subsections.

### 3.3.1 Network Architecture

The network is composed of three elements: the backbone, neck, and head. It can be formulated as,

$$\mathcal{T} = \mathcal{F}(\mathcal{S}, W) = \mathcal{G}_h(\mathcal{G}_n(\mathcal{G}_b(\mathcal{S}, W_b), W_n), W_h),$$

where  $\mathcal{G}_b$ ,  $\mathcal{G}_n$ , and  $\mathcal{G}_h$  stand for the backbone, neck, and head components respectively, while  $W_b$ ,  $W_n$ , and  $W_h$  symbolize their corresponding weights. Previous studies [44] highlighted the neck structure’s critical role in feature fusion and representation learning for detection tasks. Consequently, we introduce the Dynamic Receptive Field FPN (DRFPN), which employs a learnable receptive field approach for enhanced feature fusion. To benchmark against the current state-of-the-art (SOTA), we apply the same settings for  $\mathcal{G}_n$ ,  $\mathcal{G}_h$ , and StreamYOLO [108], leveraging CSPDarknet-53 [29] and TAL-Head [108] to build the network. Given the proven efficacy of long-term temporal information by the existing LongShortNet [51], we also integrate a dual-path architectural module for spatial-temporal feature extraction.

**Dynamic Receptive Field FPN.** Recent object detection studies, including StreamYOLO [108] and LongShortNet [51], have utilized YOLOX as their fundamental detector. YOLOX’s limitation is its fixed spatial receptive field that cannot synchronize features temporally, thus impacting its performance. To address this, we propose the Dynamic Receptive Field FPN (DRFPN) with a learnable receptive field strategy and an optimized fusion mechanism.

Specifically, Fig.3.2 contrasts PAFPN and DRFPN. PAFPN employs sequential top-down and bottom-up fusion operations to amplify feature representation. However, conventional convolution with a static kernel size fails to align features effectively. As a solution, we amalgamate the DRM module and Bottom-up Auxiliary Connect (BuAC) with PAFPN to create DRFPN. We introduce three notable modifications compared to PAFPN’s CSP module (Fig.3.2):(1) We integrate deformable convolution layers into the DRFPN module to provide the network with learnable receptive fields;(2) To enhance feature representation, we adopt re-parameterized convolutional layers [20];(3) ELAN [97] and Bottom-up Auxiliary Connect bridge the semantic gap between low and high-level features, ensuring effective detection of objects at diverse scales.

**Dual-Path Architecture.** The existing StreamYOLO [108] relies on a single historical frame in conjunction with the current frame to learn short-term motion consistency. While this suffices for ideal uniform linear motion, it falls short in handling complex motion, such as non-uniform motion (e.g., accelerating vehicles), non-linear motion (e.g., rotation of objects or camera), and scene occlusions (e.g., billboard or oncoming car occlusion).

To remedy this, we integrate the dual-path architecture [51] with a reimagined base detector, enabling the capture of long-term temporal motion while calibrating it with short-term spatial semantics. The original backbone and neck can be represented formally



as,

$$\begin{aligned}
& \mathcal{G}_n(\mathcal{G}_b(\mathcal{S}, W_b), W_n) \\
&= \mathcal{G}_{n+b}(\mathcal{S}, W_{n+b}) \\
&= \mathcal{G}_{fuse}(\mathcal{G}_{n+b}^{short}(I_t), \mathcal{G}_{n+b}^{long}(I_{t-\delta t}, \dots, I_{t-N\delta t})),
\end{aligned}$$

where  $\mathcal{G}_{fuse}$  represents the LSFM-Lf-Dil of LongShortNet.  $\mathcal{G}_{n+b}^{short}$  and  $\mathcal{G}_{n+b}^{long}$  denote the ShortPath and LongPath of LongShortNet, which are used for feature extraction of the current and historical feature, respectively. Note that their weights are shared.

Finally, the dual-path network is formulated as,

$$\begin{aligned}
\mathcal{T} &= \mathcal{F}(\mathcal{S}, W) \\
&= \mathcal{G}_h(\mathcal{G}_n(\mathcal{G}_b(\mathcal{S}, W_b), W_n), W_h) \\
&= \mathcal{G}_h(\mathcal{G}_{fuse}(\mathcal{G}_{n+b}^{short}(I_t), \mathcal{G}_{n+b}^{long}(I_{t-\delta t}, \dots, I_{t-N\delta t}))),
\end{aligned}$$

where the proposed dual-path architecture effectively addresses complex motion scenarios and offers a sophisticated solution for object detection in video sequences.

### 3.3.2 Asymmetric Knowledge Distillation

The ability to retain long-term spatiotemporal knowledge through fused features lends strength to forecasting, yet achieving streaming perception remains a daunting task. Drawing inspiration from knowledge distillation, we’ve fashioned an asymmetric distillation strategy, transferring “future knowledge” to the present frame. This assists the model in honing its accuracy in streaming perception without the burden of additional inference costs.

Given the asymmetric input nature of the teacher and student networks, a sizable gap emerges in their feature distributions, thus impairing the effectiveness of distillation at the feature level. Logits-based distillation primarily garners performance improvements by harmonizing the teacher model’s response-based knowledge, which aligns knowledge distribution at the semantic level. This simplifies the optimization process for asymmetric distillation. As a result, we’ve engineered a distillation module to convey rich semantic and localization knowledge from the teacher (the future) to the student (the present).

The asymmetric distillation is depicted in Fig. 3.4. The teacher model is a still image detector that takes  $I_{t+1}$  as input and produces logits for  $I_{t+1}$ . The student model is a standard streaming perception pipeline that uses historical frames  $I_{t-1}, \dots, I_{t-N}$  and the current frame  $I_t$  as input to forecast the results of the arriving frame  $I_{t+1}$ . The logits produced by the teacher and student are represented by  $\mathcal{T}_{stu} = \{F_{stu}^{cls}, F_{stu}^{reg}, F_{stu}^{obj}\}$ , and

$\mathcal{T}_{tea} = \{F_{tea}^{cls}, F_{tea}^{reg}, F_{tea}^{obj}\}$ , where  $F^{cls}$ ,  $F^{reg}$ , and  $F^{obj}$  correspond to the classification, objectness, and regression logits features, respectively. The Asymmetric Knowledge Distillation, AKDM( $\cdot$ ), is mathematically formulated as,

$$\begin{aligned} & \text{AKDM}(\mathcal{T}_{stu}, \mathcal{T}_{tea}) \\ &= \mathcal{L}_{cls}(F_{stu}^{cls}, F_{tea}^{cls}) + \mathcal{L}_{obj}(F_{stu}^{obj}, F_{tea}^{obj}) + \mathcal{L}_{reg}(\hat{F}_{stu}^{reg}, \hat{F}_{tea}^{reg}), \end{aligned}$$

where  $\mathcal{L}_{cls}(\cdot)$  and  $\mathcal{L}_{obj}(\cdot)$  are Mean Square Error (MSE) loss functions, and  $\mathcal{L}_{reg}(\cdot)$  is the GIoU loss [82].  $\hat{F}_{stu}^{reg}$  and  $\hat{F}_{tea}^{reg}$  represent the positive samples of the regression logit features, filtered using the OTA assignment method as in YOLOX [29]. It is worth noting that location knowledge distillation is only performed on positive samples to avoid noise from negative ones.

### 3.3.3 K-step Streaming Metric

The Streaming Average Precision (sAP) metric is a prevalent tool used to gauge the precision of Streaming Perception systems [52]. This metric gauges precision by juxtaposing real-world ground truth with system-generated results, factoring in process latency.

Two primary methodologies exist in this domain: non-real-time and real-time. For non-real-time methods, as depicted in Fig.3.3(a), the sAP metric calculates precision by comparing the current frame  $I_t$  results with the ground truth of the following frame  $I_{t+2}$ , post processing of frame  $I_t$ . Conversely, real-time methods, as demonstrated in Fig. 3.3(b), conclude the processing of the current frame  $I_t$  prior to the next frame  $I_{t+1}$  arrival. Our proposed method, DAMO-StreamNet, is a real-time method, adhering to the pipeline outlined in Fig. 3.3(b).

Though the sAP metric effectively evaluates the short-term forecasting capability of algorithms, it falls short in assessing their long-term forecasting prowess—a critical factor in real-world autonomous driving scenarios. In response, we introduce the K-step Streaming metric, an expansion of the sAP metric, specifically tailored to evaluate long-term performance. As depicted in Fig. 3.3(c), the algorithm projects the results of the upcoming two frames, and the cycle continues. The projection of the next K frames is represented as "K-sAP", as shown in Fig. 3.3(d). Consequently, the standard sAP metric translates to 1-sAP in the K-step metric context.

## 3.4 LONGSHORTNET

We argue that *spatial semantics* and *temporal motion* are crucial for detecting complex movements such as non-uniform, non-linear, and occlusion. To this end, we propose *LongShortNet*, which coherently models long-term temporal and fuses it with short-term

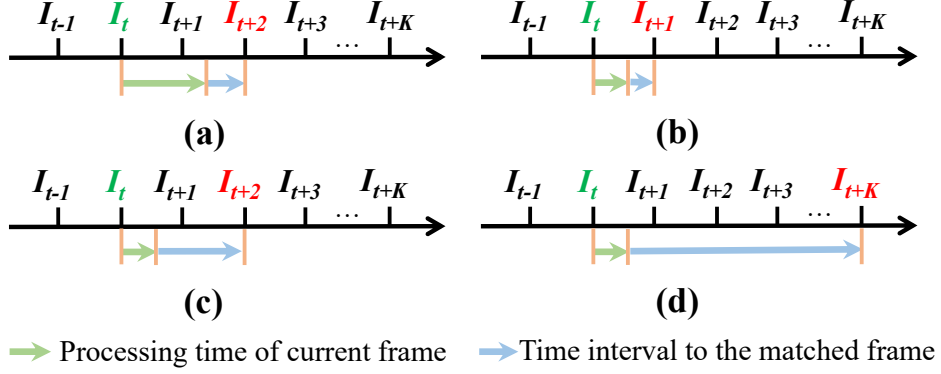


Figure 3.3: Illustration of matching rules under different metrics. The frames in **green** font denote the current frame and the frames in **red** font denote the frames matched with the current frame under the specific metric. (a) Matching result of non-real-time methods under 1-sAP. (b) Matching result of real-time methods under 1-sAP. (c) Matching result of real-time methods under 2-sAP. (d) Matching result of real-time methods under K-sAP.

semantics. LongShortNet consists of *ShortPath* and *LongPath*, as shown in Fig. 3.4(a). The frame sequence captured by the camera is divided into the current frame and the support frame, which are fed into ShortPath and LongPath, respectively, to generate spatial and temporal features. The Long-Short Fusion Module (*LSFM*) aggregates short-term spatial and long-term temporal information to capture motion consistency for representation learning. Finally, a detection *head* predicts upcoming results based on the features produced by the LSFM.

Formally, ShortPath takes the current frame  $I_t$  as input and outputs spatial features  $F_t = \mathcal{F}(I_t)$ , where  $\mathcal{F}(\cdot)$  is CNN networks, which includes the backbone (CSPDarknet-53 [6]) and the neck (PANet [60]). Similarly, LongPath stores temporal features  $F_{t-i\delta t} = \mathcal{F}(I_{t-i\delta t})$ ,  $i \in [1, N]$  where  $N$  and  $\delta t$  denote the number of frames and time steps, respectively.  $\mathcal{F}(\cdot)$  represents the network of LongPath. Note that the backbone of Short/Long paths is weight-shared. By introducing tunable parameters  $N$  and  $\delta t$ , LongPath can capture more long-term temporal for fine movement reasoning. Then LSFM aggregates all features through  $F_{fuse} = \text{LSFM}(F_t, \dots, F_{t-N\delta t})$ , where  $F_{fuse}$  denotes the fused features generated by LSFM. The details of LSFM( $\cdot$ ) are described in the next section. Finally, the results are acquired by  $D_{res} = \mathcal{H}(F_{fuse})$ , where  $\mathcal{H}$  denotes the detection head (TALHead [108]) and  $D_{res}$  are predicted locations, scores, and categories.

### 3.4.1 Long Short Fusion Module

The previous streaming perception work [108] only *roughly concatenates the features of the last two frames*, without exploiting temporal motion and spatial semantics. We investigate a variety of feature aggregation ways, including 1) early fusion vs. late fusion and 2) average (equal weights) vs. dilatation (different weights). In summary, we verified

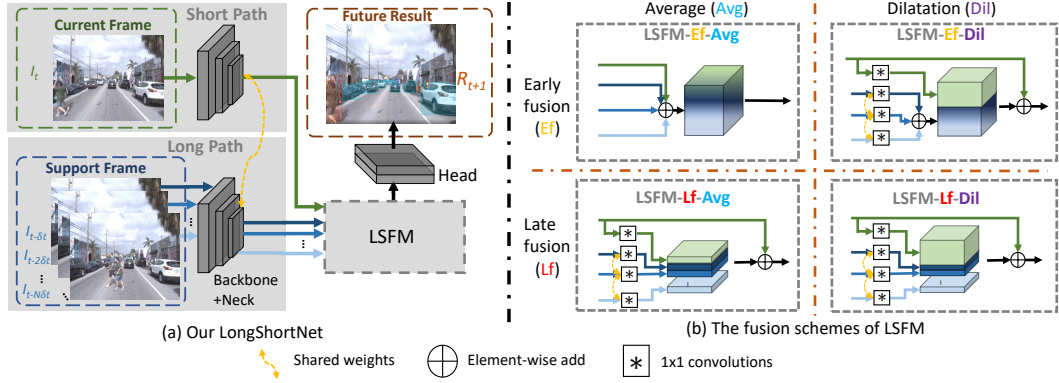


Figure 3.4: Illustration of our LongShortNet. (a) is an overview of LongShortNet. (b) shows the details of different fusion schemes of LSFM.

four types of LSFM as shown in Fig. 3.4(b), denoted as LSFM-Ef-Avg, LSFM-Ef-Dil, LSFM-Lf-Avg, and LSFM-Lf-Dil.

**Average-Early-Fusion.** The LSFM-Ef-Avg process fuses the spatial semantics of each frame in LSFM and outputs pre-averaged synthetic spatiotemporal features for the detection head. This vanilla version allocates equal importance to the features of all frames, which is defined as,

$$F_{fuse} = \sum_{i=1}^N F_{t-i\delta t} + F_t, \quad (3.1)$$

where it counts all the features to fuse the current/historical spatial information directly and equally.

**Dilatation-Early-Fusion.** For LSFM-Ef-Dil, we investigate different weighting schemes for feature fusion as,

$$F_{fuse} = \text{Concat}(\mathcal{G}_{short}(F_t), \sum_{i=1}^N \mathcal{G}_{long}(F_{t-i\delta t})) + F_t, \quad (3.2)$$

where  $\mathcal{G}$  denotes the  $1 \times 1$  convolution operation and Concat means the channel-wise concatenation. Supposed that the channel dimensionality of  $F_t$  and  $F_{t-i\delta t}$  is  $d$ , all long-term temporal features are fused by addition before concatenating with the short-term spatial features. In this case, the output channels numbers of  $\mathcal{G}_{short}(\cdot)$  and  $\mathcal{G}_{long}(\cdot)$  are both  $\lfloor d/2 \rfloor$ . Note that we also adopt a residual connection to add current spatial features to enhance the historical temporal features.

**Average-Late-Fusion.** Contrary to the early fusion, LSFM-Lf-Avg fusion preserves the spatial semantic features of each frame separately and relies on the detection head to extract more high-level coherent features. It instantly concatenates all features without discriminating between ShortPath and LongPath, which is defined as,

$$F_{fuse} = \text{Concat}(\mathcal{G}_{avg}(F_t), \dots, \mathcal{G}_{avg}(F_{t-N\delta t})) + F_t, \quad (3.3)$$

where the output channels number of  $\mathcal{G}_{avg}(\cdot)$  is  $\lfloor d/(1+N) \rfloor$ . LSFM-Lf-Avg treats all features equally.

**Dilatation-Late-Fusion.** We further propose LSFM-Lf-Dil, which enlarges the number of channels of ShortPath and forces LongShortNet to pay more attention to the current spatial information. Specifically, LSFM-Lf-Dil is defined as,

$$F_{fuse} = \text{Concat}(\mathcal{G}_{short}(F_t), \dots, \mathcal{G}_{long}(F_{t-N\delta t})) + F_t, \quad (3.4)$$

where two  $1 \times 1$  convolution operations are employed to project  $F_t$  and  $F_{t-i\delta t}$  separately. The output channels numbers of  $\mathcal{G}_{short}(\cdot)$  and  $\mathcal{G}_{long}(\cdot)$  are  $\lfloor d/2 \rfloor$  and  $\lfloor d/2N \rfloor$ . After extensive experimental comparison, we finally chose Dilatation-Late-Fusion as LSFM and set  $N$  and  $\delta t$  to 3 and 1.

Table 3.1: Comparison with both non-real-time and real-time state-of-the-art (SOTA) methods on the Argoverse-HD benchmark dataset. The symbol ‘ $\ddagger$ ’ denotes the use of a large size (1200, 1920) and extra data. The symbol ‘ $\dagger$ ’ denotes the use of a large size (1200, 1920) without the use of extra data. The best results for each setting are shown in **green**. The largest increments of the large resolution setting are shown in **red**.

Methods	sAP	sAP <sub>50</sub>	sAP <sub>75</sub>	sAP <sub>s</sub>	sAP <sub>m</sub>	sAP <sub>l</sub>
Non-real-time detector-based methods						
Streamer (S=900) [52]	18.2	35.3	16.8	<b>4.7</b>	14.4	34.6
Streamer (S=600) [52]	20.4	35.6	20.8	3.6	18.0	47.2
Streamer + AdaScale [13, 32]	13.8	23.4	14.2	0.2	9.0	39.9
Adaptive Streamer [32]	<b>21.3</b>	<b>37.3</b>	<b>21.1</b>	4.4	<b>18.7</b>	<b>47.1</b>
Real-time detector-based methods						
StreamYOLO-S [108]	28.8	50.3	27.6	9.7	30.7	53.1
StreamYOLO-M [108]	32.9	54.0	32.5	12.4	34.8	58.1
StreamYOLO-L [108]	36.1	57.6	35.6	13.8	37.1	63.3
LongShortNet-S (Ours) [51]	29.8	50.4	29.5	11.0	30.6	52.8
LongShortNet-M (Ours) [51]	34.1	54.8	34.6	13.3	35.3	58.1
LongShortNet-L (Ours) [51]	37.1	57.8	37.7	15.2	37.3	63.8
DAMO-StreamNetNet-S (Ours) [34]	31.8	52.3	31.0	11.4	32.9	58.7
DAMO-StreamNetNet-M (Ours) [34]	35.7	56.7	35.9	14.5	36.3	63.3
DAMO-StreamNetNet-L (Ours) [34]	<b>37.8</b>	<b>59.1</b>	<b>38.6</b>	<b>16.1</b>	<b>39.0</b>	<b>64.6</b>
Large resolution						
StreamYOLO-L $\ddagger$	41.6	65.2	43.8	23.1	44.7	60.5
LongShortNet-L (Ours) [51] $\dagger$	42.7 (+1.1)	65.4 (+0.2)	<b>45.0 (+1.2)</b>	23.9 (+0.8)	44.8 (+0.1)	61.7 (+1.2)
DAMO-StreamNet-L $\dagger$ (Ours) [34]	<b>43.3 (+1.7)</b>	<b>66.1 (+0.9)</b>	44.6 (+0.8)	<b>24.2 (+1.1)</b>	<b>47.3 (+2.6)</b>	<b>64.1 (+3.6)</b>

## 3.5 EXPERIMENTS

### 3.5.1 Dataset and Metric

**Dataset:** We utilized the Argoverse-HD dataset, which comprises various urban outdoor scenes from two US cities. The dataset contains detection annotations and center RGB camera images, which were used in our experiments. We adhered to the train/validation split proposed by Li et al. [52], with the validation set consisting of 15k frames.

**Evaluation Metrics:** We employed the streaming Average Precision (sAP) metric to evaluate performance. The sAP metric calculates the average mAP over Intersection over Union (IoU) thresholds ranging from 0.5 to 0.95, as well as APs, APm, and AP<sub>l</sub>

for small, medium, and large objects, respectively. This metric has been widely used in object detection, including in previous works such as [52, 108].

### 3.5.2 Implementation Details

Our DAMO-StreamNet and LongShortNet models are both premised upon the YOLOX base detector [29], initially pretrained on the COCO dataset [56]. We further fine-tuned these frameworks on the Argoverse-HD dataset for a total of 8 epochs, deploying a batch size of 32 and 16 respectively, with the aid of 4 V100 GPUs. Both models were developed in three configurations: small, medium, and large, named DAMO-StreamNet-S/M/L and LongShortNet-S/M/L, designed with the intent to facilitate convenient comparison with recent state-of-the-art models [108, 51]. The standard input resolution (600, 960) was maintained unless otherwise indicated. Hyperparameters were chosen consistently with previous works [108, 51] to ensure fair comparison. In the case of DAMO-StreamNet, the AK-Distillation served as an auxiliary loss during the training process, with the loss weight set at 0.2/0.2/0.1 for the small, medium, and large models respectively. To guarantee the real-time performance of the network, we adopted and made minor adjustments to the buffer scheme proposed in [108].

### 3.5.3 Comparison with State-of-the-art Methods

We compared our proposed approach with state-of-the-art methods to evaluate its performance. In this subsection, we directly copied the reported performance from their original papers as their results. The performance comparison was conducted on the Argoverse-HD dataset [52]. An overview of the results reveals that our proposed DAMO-StreamNet with an input resolution of  $600 \times 960$  achieves 37.8% sAP, outperforming the current state-of-the-art methods by a significant margin. For the large-resolution input of  $1200 \times 1920$ , our DAMO-StreamNet attains 43.3% sAP without extra training data, surpassing the state-of-the-art work StreamYOLO, which was trained with large-scale auxiliary datasets. This clearly demonstrates the effectiveness of the systematic improvements in DAMO-StreamNet.

Compared to StreamYOLO and LongShortNet, DAMO-StreamNet-L achieves absolute improvements of 3.6% and 2.4% under the sAP<sub>L</sub> metric, respectively. This also provides substantial evidence that the features produced by DRFPN offer a self-adaptive and sufficient size of the receptive field for large-sized objects. It is worth noting that DAMO-StreamNet experiences a slight decline compared to LongShortNet under the stricter metric sAP<sub>75</sub>. This observation suggests that although the dynamic receptive field achieves a sufficient receptive field for different scales of objects, it is not as accurate as fixed kernel-size ConvNets. The offset prediction in the deformable convolution

layer may not be precise enough for high-precision scenarios. In other words, better performance could be achieved if this issue is addressed, and we leave this for future work.

Table 3.2: Ablation study of the base detector on the Argoverse-HD dataset. The best results for each subset and the corresponding increments are shown in green font and red font, respectively.

Methods	S	M	L
Equip StreamYOLO with our DRFPN			
StreamYOLO	28.7	33.5	36.1
+DRFPN	<b>30.6 (+1.9)</b>	<b>35.1 (+1.6)</b>	<b>36.7 (+0.6)</b>
LongShortNet Equipped with our DRFPN			
LongShortNet	29.8	34.0	36.7
+DRFPN	<b>31.5 (+1.7)</b>	<b>35.7 (+1.7)</b>	<b>37.5 (+0.8)</b>

### 3.5.4 Ablation Study

**Investigation of DRFPN.** To verify the effectiveness of DRFPN, we use StreamYOLO [108] and LongShortNet [51] as baselines and integrate them with the proposed DRFPN, respectively. The experimental results are listed in Table 3.2. It is evident that DRFPN significantly improves the feature aggregation capability of the baselines. Particularly, the small-scale baseline models equipped with DRFPN achieve improvements of 1.9% and 1.7%, separately. This also demonstrates that the dynamic receptive field is crucial for the stream perception task. More importantly, DRFPN enhances the performance of LongShortNet, which suggests that the temporal feature alignment capacity is also augmented by the dynamic receptive field mechanism.

Table 3.3: Exploration of  $N$  and  $\delta t$  on the Argoverse-HD dataset. StreamNet denotes our DAMO-StreamNet. The best two results and the worst one are shown in green font, blue font, and purple font, respectively. The best increments are shown in red font.

$(N, \delta t)$	StreamNet-S	StreamNet-M	StreamNet-L
(0, -)	<b>28.1</b>	<b>32.0</b>	<b>34.2</b>
(1, 1)	30.6	35.1	36.7
(1, 2)	31.2	34.5	37.1
(2, 1)	31.2	<b>35.7 (+3.7)</b>	<b>37.5 (+3.3)</b>
(2, 2)	<b>31.4 (+3.3)</b>	<b>35.4 (+3.4)</b>	37.2
(3, 1)	<b>31.5 (+3.4)</b>	35.3	37.2
(3, 2)	31.2	35.1	<b>37.4 (+3.2)</b>
(4, 1)	31.1	35.0	37.1
(4, 2)	30.7	35.2	36.5
(5, 1)	31.1	35.0	<b>37.5 (+3.3)</b>
(5, 2)	30.9	34.7	36.9

**Investigation of Temporal Range.** To isolate the influence of temporal range, we conduct an ablation study on  $N$  and  $\delta t$ , as listed in Table 3.3. (0, -) represents the model utilizing only the current frame as input. It is evident that increasing the number of input frames can enhance the model’s performance, with the best results obtained when  $N$  is equal to 2, 2, and 3 for DAMO-StreamNet-S/M/L, respectively. However, as the number of input frames continues to increase, the performance experiences significant declines. Intuitively, longer temporal information should be more conducive to forecasting, but the effective utilization of long-term temporal information remains a critical challenge worth investigating.

Table 3.4: Ablation study of our proposed models. D-SN and AK-D represent DAMO-StreamNet and AK-Distillation, respectively. The best results and the largest increments are shown in green font and red font, respectively.

Methods	S	M	L
D-SN (N=1)	30.6	35.1	36.7
D-SN (N=1)+AK-D	31.5 (+0.9)	35.3 (+0.2)	37.1 (+0.4)
D-SN (N=2/3)	31.5	35.7	37.5
D-SN (N=2/3)+AK-D	31.8 (+0.3)	35.5 (-0.2)	37.8 (+0.3)

**Investigation of AK-Distillation.** AK-Distillation is a cost-free approach for enhancing the streaming perception pipeline, and we examine its impact. We perform AK-Distillation with various lengths of temporal modeling and scales of DAMO-StreamNet. As the results listed in Table 3.4 indicate, AK-Distillation yields improvements of 0.2% to 0.9% for the DAMO-StreamNet configured with  $N = 1$  short-term temporal modeling. This demonstrates that AK-Distillation can effectively transfer ”future knowledge” from the teacher to the student. For the DAMO-StreamNet with the setting of  $N = 3$ , AK-Distillation improves DAMO-StreamNet-S/L by only 0.3%, but results in a slight decline for the medium-scale model. The limited improvement for long-term DAMO-StreamNet is due to the narrow performance gap between the teacher and student, and the relatively high precision is difficult to further enhance.

**Investigation of K-step Streaming Metric.** We evaluate DAMO-StreamNet with settings  $N = 1$  and  $N = 2/3$  under the new metric  $sAP_k$ , where  $k$  ranges from 1 to 6. The results are listed in Table 3.5. It is clear that the performance progressively declines as  $k$  increases, which also highlights the challenge of long-term forecasting. Another observation is that the longer time-series information leads to better performance under the new metric.

**Inference Efficiency Analysis.** Although the proposed DRFPN has a more complex structure compared to PAFPN, DAMO-StreamNet still maintains real-time streaming perception capabilities. For long-term fusion, we adopt the buffer mechanism from



Table 3.5: Exploration study of K-sAP on the Argoverse-HD dataset. Here, our proposed model DAMO-StreamNet is denoted as StreamNet. The best results and largest increments for each subset are shown in green and red font, respectively.

K-Step Metric		StreamNet (N=1)	StreamNet (N=2/3)
S	sAP <sub>1</sub>	30.6	<b>31.5</b> (+0.9)
	sAP <sub>2</sub>	28.3	29.8 ( <b>+1.5</b> )
	sAP <sub>3</sub>	24.9	25.9 (+1.0)
	sAP <sub>4</sub>	22.1	23.3 (+1.2)
	sAP <sub>5</sub>	21.0	21.8 (+0.8)
	sAP <sub>6</sub>	18.8	20.0 (+1.2)
M	sAP <sub>1</sub>	35.1	<b>35.7</b> (+0.6)
	sAP <sub>2</sub>	31.9	32.8 ( <b>+0.9</b> )
	sAP <sub>3</sub>	28.8	29.2 (+0.4)
	sAP <sub>4</sub>	25.7	25.9 (+0.2)
	sAP <sub>5</sub>	23.2	23.4 (+0.2)
	sAP <sub>6</sub>	21.5	22.0 (+0.5)
L	sAP <sub>1</sub>	36.7	<b>37.5</b> ( <b>+0.8</b> )
	sAP <sub>2</sub>	33.2	33.9 (+0.7)
	sAP <sub>3</sub>	29.8	30.6 ( <b>+0.8</b> )
	sAP <sub>4</sub>	27.1	27.2 (+0.1)
	sAP <sub>5</sub>	24.2	25.0 ( <b>+0.8</b> )
	sAP <sub>6</sub>	22.3	22.7 (+0.4)

StreamYOLO [108], which incurs only minimal additional computational cost for multi-frame feature fusion.

Table 3.6: Ablation study of inference time (ms) on V100.

Methods	S	M	L
LongShortNet (N=1)	14.2	17.3	19.7
LongShortNet (N=3)	14.6	17.5	19.8
DAMO-StreamNet (N=1)	21.0	24.2	26.2
DAMO-StreamNet (N=3)	21.3	24.3	26.6

### 3.6 CONCLUSION & FUTURE WORK

We introduced DAMO-StreamNet and LongShortNet exhibit several key enhancements: (1) a fortified neck structure with deformable convolution; (2) a dual-branch structure for deeper time-series data analysis; (3) logits layer distillation for improved deep learning model interpretation; and (4) a cutting-edge real-time forecasting mechanism that perpetually updates frame features. Evaluation on the Argoverse-HD dataset underscores our models’ superiority over their state-of-the-art peers.

In future work, we intend to: (1) incorporate explicit motion consistency constraints based on geometric context to improve performance robustness and accuracy in complex settings; and (2) extend these methods’ applications in autonomous systems areas such as planning and control, potentially advancing autonomous driving technology.

# CHAPTER 4

## VEHICLE TRACKING USING NATURAL LANGUAGE DESCRIPTIONS

### 4.1 INTRODUCTION

The AI-CITY challenge [72] focuses on the development of intelligent traffic systems. Vehicle tracking refers to the task of retrieving a *track* given an input query. A *track* refers to the camera id the vehicle is observed in, the frame number & corresponding location coordinates (bounding boxes) of the object within the frame. Track 2 of the AI-CITY challenge focuses on a specific task where in the input data, natural language description of the vehicle are also provided eg. “A red sedan turns right at the intersection”.

This introduces a new challenge of combining vision with natural language techniques for accurate vehicle retrieval. Existing works have focused on using a dual-stream architecture utilizing language encoders and visual encoders to obtain feature representations from both modalities and utilize a form of contrastive learning to effectively predict output tracks. Prior works [53, 76, 71] have also looked to find better ways to represent *static* properties such as vehicle color, type, shape and *dynamic* properties such as motion, speed and position relative to other vehicles.

In this paper, we propose to explore some research questions related to the contribution of different features to vehicle tracking. The key research questions for this paper as follows:

- What is the contribution of static and dynamic properties to model performance?
- How much does the language modality contribute to model performance?
- To what extent can language features help capture both static and dynamic properties?
- How well can enhanced visual/video transformers capture tracking-features?

To answer these research questions, we conduct our study based on four major experiments. The first experiment deals with data augmentation to find the extent language contributes to the model performance. Our results show that augmenting the natural language descriptions make the model more robust and perform better on the retrieval task. We also add a prompt feature to explicitly pass *color* and *type* features of

the vehicle to the model. We engineer a manual template to obtain this prompt feature and find that although our proposed template performs better than existing prompt-based approaches, it does not contribute to model performance significantly.

We also build a modified symmetric network to obtain *motion* features from the language description. We notice significant improvement in model performance with the presence of motion features, showing the importance of dynamic properties for vehicle retrieval. We also replace existing visual encoders and utilize SWIN transformers for better and efficient visual embeddings. Our model shows comparable performance to existing systems. We also perform an ablation study to show the contributions of different modalities and features.

## 4.2 RELATED WORK

### 4.2.1 Connecting Language and Vision

[3] propose to jointly train the vision and transformer-based language model using symmetric InfoNCE loss and instance loss. As shown in Figure 4.1, the vanilla network consists of (1) local image encoder for local cropped vehicle image, (2) text encoder for language description input, (3) global image encoder to help learning more position and motion information.

The work proposes to augment visual training data especially to capture the global/external features. The method proposes fixing a background and then capturing the different vehicle bounding boxes at different timestamps to form a “motion image”.

The work deploys a “dual-stream” structure using local images and global “motion” images. The local images are the detected vehicles, cropped from a random frame. The global motion image is obtained using the method specified in the paragraph above. The dual stream structures involves two independent CNN encoders pre-trained on ImageNet.

For each stream, they introduce 3 projection heads (local features, global motion features and the concatenated fusion feature) to map visual representation to space of contrastive representation learning. Additionally, classification heads output the predicted probability of different tracks (similar to projection head, but the output dimension is the number of tracks).

For text embeddings, the authors deploy BERT or RoBERTa as text encoder. The projection head maps text embeddings to the space of contrastive representation learning.

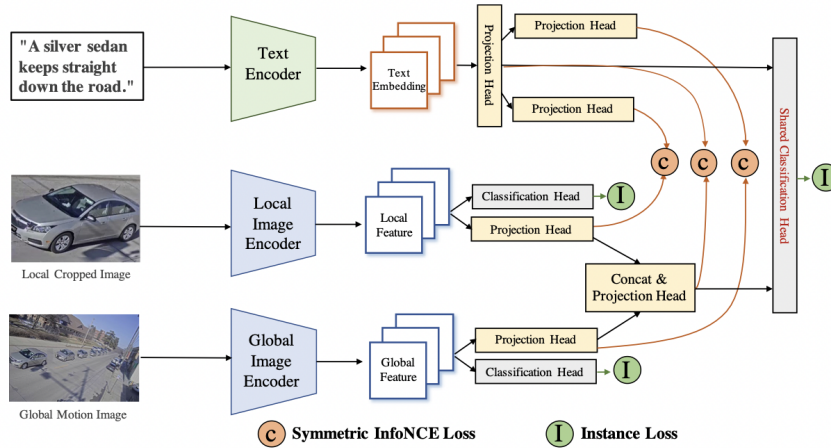


Figure 4.1: Jointly train vision, language model

#### 4.2.2 Symmetric Network

[116] builds upon the previous work by using a symmetric network to encode visual and language features. The work proposes similar image augmentation method. They improve generating "motion images" when the vehicle is too close in adjacent frames. By calculating the IOU (Intersection over Union) of adjacent frames, the paper proposes to discard frames larger than a threshold IOU.

For language augmentation, the authors propose first to use spacy to extract noun phrases and pick the first one and append it to the beginning. The idea is based on the assumption that the first noun is usually a description of the vehicle and the appearance can be enhanced by repeating the description.

As shown in Figure 4.2, the proposed approach is to use separate visual encoders for local images and for global images. Both will also have a text encoder to enhance information of either local or global image with features from the textual descriptions.

For the local and global image encoders, the cropped image of the vehicle or the motion image is fed as input to Efficient Net B2 [107] or ibn-ResNet101-a [75] pretrained on ImageNet. For text encoding, they use RoBERTa [63].

They propose to concatenate the local and global image features to fuse information at different granularities. The fused visual and textual representations are projected into the same space by the projection heads. They also use InfoNCE loss.

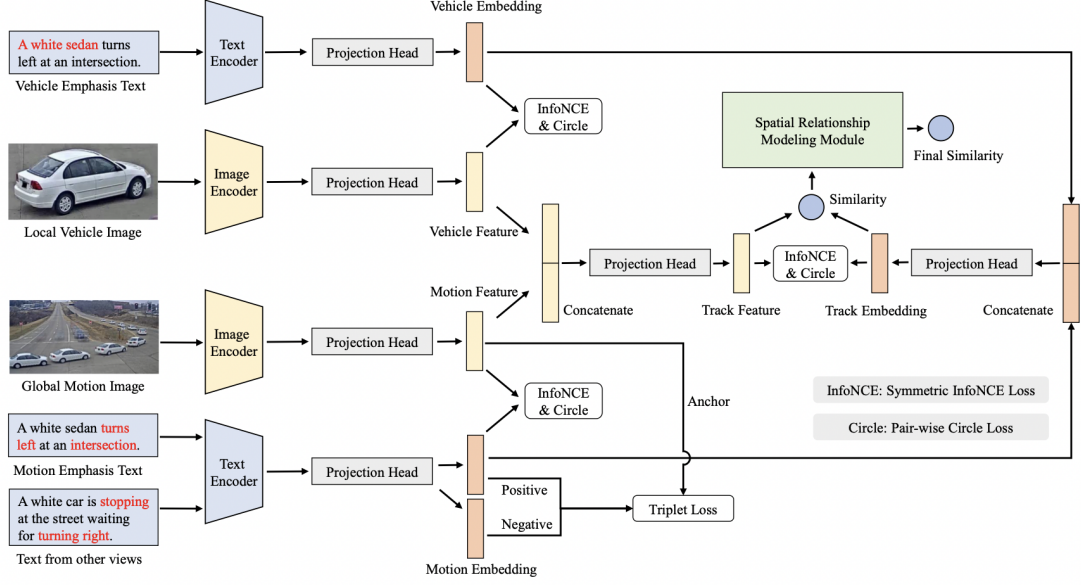


Figure 4.2: Symmetric model with language and vision encoders for static and dynamic properties separately

### 4.2.3 Multi-granularity Retrieval System

[115] propose a multi-granular system with 3 main modules: (1) Language parsing: to obtain attributes of the vehicle from the language descriptions (2) Language-augmented multi-query: vehicle track retrieval module that serves as baseline model to incorporate information from multiple imperfect queries (3) Target vehicle attributes enhancement module: which explicitly fuses the static and dynamic properties of the target vehicle to generate final retrieval results.

The authors note that most language queries have a similar language structure: *main subject + action + (optional other subject + action)*. Hence they employ Semantic Role Labeling technique and words-frequency voting to collect main attribute words from the query which can help define characteristics of the car  $L_{color}$ ,  $L_{type}$ , and  $L_{direction}$ .

This work proposes using BaiduNLP to augment the query set to include more imperfect sentences. A sample  $N_q$  of this augmented set of queries  $Q_i$ , vehicle track images  $v_i$  and motion images  $m_i$  are passed as inputs of a multi-query vehicle tracking model. The Motion  $E_m(\cdot)$  and Vehicle Track  $E_v(\cdot)$  encoder use the Spatial-Temporal Transformer Encoder [4], producing corresponding vehicle track feature  $f_{track}$  and motion feature  $f_{motion}$ .

$f_{track}$  is forwarded through two different fully connected layers to extract the color embedding  $f_{color}$  and the type embedding  $f_{type}$ , which are learned using labels  $L_{color}$  and  $L_{type}$ . Furthermore a re-identification feature extractor [73] is used to obtain  $f_{reid}$

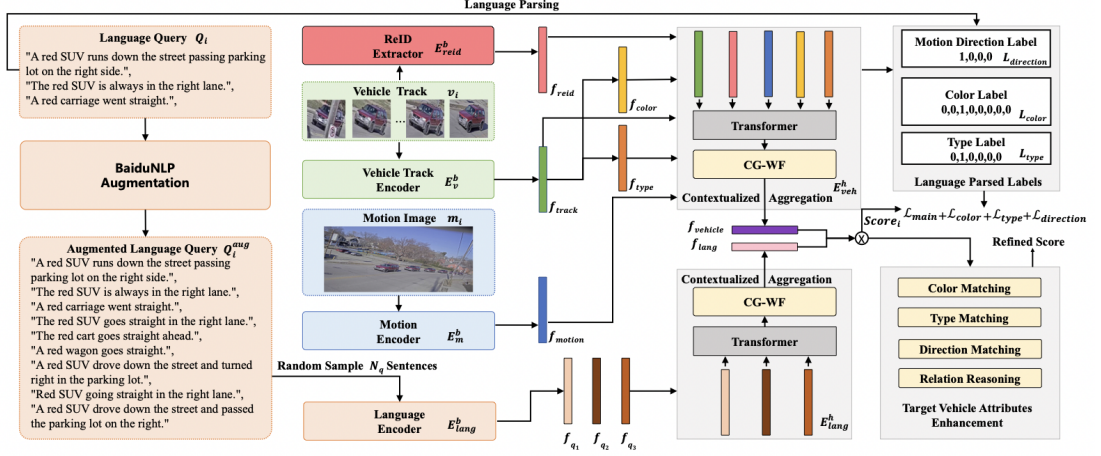


Figure 4.3: Multi granular model with post processing module for reweighting text-language match scores

Model	MRR%
Connecting Vision & Language	18.69
Symmetric Model	43.92
Multi-granular Retrieval	56.52

Table 4.1: Baseline MRR performance

from the vehicle images  $v_i$ .

A contextualized aggregation network, based on transformer attention network  $E_{veh}$  is used to obtain

$$f_{vehicle} = E_{veh}([f_{track}, f_{color}, f_{type}, f_{reid}, f_{motion}])$$

The language encoder samples a set of  $n$  queries from the augmented dataset  $Q_i$

$$f_{lang} = E_{lang}([f_1, \dots, f_n])$$

where  $E_{lang}$  is distilbert model [101].

The cross-modal matching similarity is calculated by a simple dot product between  $f_{vehicle}$  and  $f_{lang}$ . The paper then goes on to describe a Target Vehicle Attribute Enhancement module to align the static and dynamic properties of the target vehicle by re-weighting the retrieval results. This in fact helps improve MRR by almost 15% which is a significant increase.

### 4.3 PROPOSED APPROACH

We now describe the several experiments we ran based on the research questions described in Section 4.1.

---

*Text:* A mid-sized black SUV drives straight down a road behind another SUV.

---

*Translation:* 一辆中型黑色SUV在另一辆SUV后面的道路上直奔。

---

*Back-translation:* A medium-sized black SUV went straight on the road behind the other.

---

Figure 4.4: Backtranslation for Data Augmentation

### 4.3.1 Data Augmentation

Natural language descriptions are an addition to the AI-CITY track 2 dataset. However, there are only 3 descriptions per query. In order for the model to obtain better understanding of the language features, we propose to replace the BaiduNLP Augmentation [115] with a simple backtranslation technique.

Backtranslation provides more training data to improve model robustness. It does so by converting a sentence in one source language (English) to another (say, Chinese) and translating it back to the source language. This generates semantic invariants for each of the natural language descriptions in the training samples. We utilize the data collected by [3]. Translating to languages that are similar to English, such as French and German, may cause backtranslation to generate the same texts. Hence, the texts are translated to Chinese and then back to English. An example of this is Figure 4.4.

Entity strengthening is used by 4.2.2 by extracting noun phrases such as “white sedan” from Spacy. They then proceed to repeat this phrase in the sentence by appending it to the beginning of the input sentence. The purpose of this is to provide more information to the language encoder about the static properties of the vehicle such as its color and type. Although we utilize this data augmentation technique, we argue that there are better methods to capture static properties of vehicle as described in the next subsection.

### 4.3.2 Prompt Tuning

Prompting has proven to add crucial information to NLP tasks [59]. [23] propose a prompt-tuning technique (Figure 4.5) to provide explicit features to the model regarding the static properties of the model such as the color and type. In order to do this, they use a dependency parser [40] to obtain noun-adjective phrases from the input sentence. Since the natural language descriptions in the dataset are of simple and consistent structure, it is easy to obtain these phrases. After that they utilize a manual prompt template “*This is a [COLOR] [TYPE]*” eg. *This is a blue SUV*. This prompt template is fed additionally to the language encoder to obtain a prompt feature to better improve model performance.

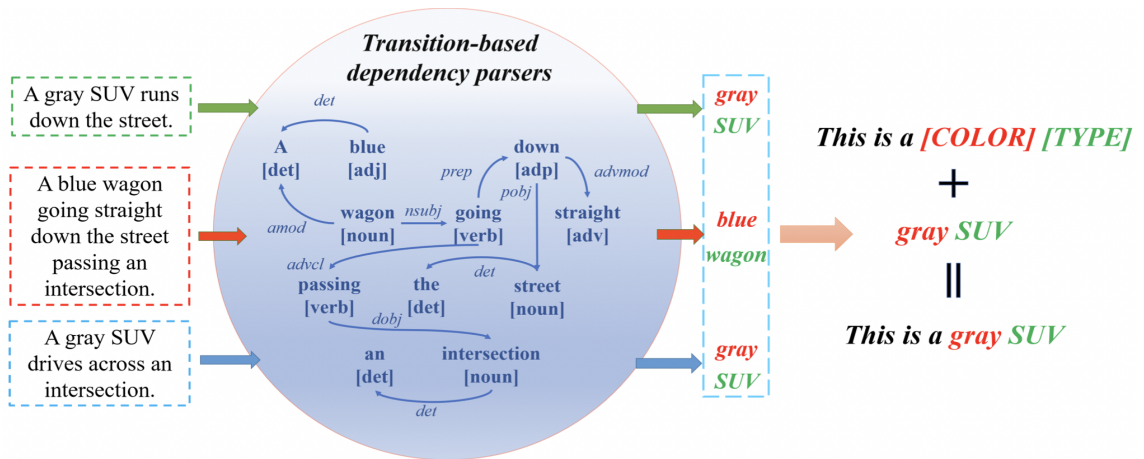


Figure 4.5: Prompt Tuning for better static property representation

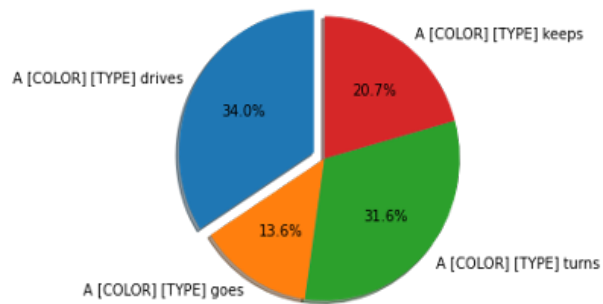


Figure 4.6: Common language patterns in Train set

Pattern	Count
A [COLOR] [TYPE] drives	110
A [COLOR] [TYPE] goes	49
A [COLOR] [TYPE] turns	113
A [COLOR] [TYPE] keeps	68

Table 4.2: Tabulated number of language templates in Validation Set

Model	MRR %
[23]	30.12
A [COLOR] [TYPE] goes	29.95
A [COLOR] [TYPE] drives	31.97

Table 4.3: Model performance with manual prompt templates



We perform an analysis on the dataset and observe that a large number of input descriptions follow a fixed language pattern. An analysis of such language patterns is shown in Figure 4.6 and Table 4.2. We observe that a large number of sentences follow a language pattern “A [COLOR] [TYPE] drives/turns” eg. “A white sedan drives down the road..” or “A white sedan turns left at an intersection”. We propose that a prompt template which mirrors such a pattern will be able to better capture the static properties of a vehicle. This is similar to the idea of entity strengthening or highlighting, but is a more intuitive way to provide information to the text encoder.

We propose to utilize the manual template “A [COLOR] [TYPE] drives” and experiment with other templates defined in Table 4.2, instead of the Manual Template proposed by [23]. We argue that this technique will help better capture the properties of the vehicle since these will be a form of data augmentation. The language descriptions already follow a similar template and providing this explicit prompt feature will help reinforce the color and type features to the model. This is backed by the performance of model with manual template as shown in Table 4.3.

We also tried to include other properties such as position and speed using prompt tuning. However, the language descriptions in the existing dataset do not provide much information to extract such features solely from the text modality. We instead need to resort to a better way to obtain such features from the text and vision modalities. What could be obtained from the language description is information about the motion (eg. “turns left”, “turns right” or “keeps straight”). We do not find an intuitive way to include this in the manual prompt and hence propose a more sophisticated technique for this in the next subsection.

### 4.3.3 Modified Symmetric Network

We build upon the Symmetric network of [116] and propose our new model architecture. The symmetric model used by the prior work focuses on utilizing one text encoder and one image encoder (for local image) to obtain static property features and another pair of one text encoder, one image encoder (for global motion image) to obtain dynamic/global property features.

Since we pass different inputs to the visual encoders, it makes sense to use such a symmetric network. We pass a static image of a car just for it to capture static/local properties such as vehicle color, type, shape, size. However, for dynamic/global properties we use a global image motion map. Hence the need to use another visual encoder to obtain features such as motion, direction, velocity. However, we argue that using two different encoders for the text modality is a complex and redundant architecture.

We propose a novel text encoding architecture to obtain static properties and the motion property. The way we can include information about the motion/direction explicitly to the text encoder is through keywords in the description eg. “turns left” or “turns right” or “keeps straight”. This is a better way to capture motion embeddings rather than using entity strengthening or other methods in previous works.

We use DistilBERT [101] to obtain three features from the input sentence descriptions  $L_{color}$ ,  $L_{type}$  and  $L_{direction}$ . These denote embeddings for color, type and motion of the vehicle respectively from the language input. We obtain similar features  $V_{color}$ ,  $V_{type}$  and  $V_{direction}$  from the visual encoders which is Efficient Net B3 [90] instead of the B2 used in the prior work. We perform a late fusion technique where we project the embeddings to the same space and average the features. We also experimented by sum-pooling the features.

To maximize the similarity between the features learnt from the visual encoders and text encoders, we use symmetric InfoNCE Loss [106]. The image to text loss is calculated using the following equation

$$\mathcal{L}_{i2t} = \frac{1}{N} \sum_{i=1}^N -\log \frac{\exp(\cos(V_{feat}^i, L_{feat}^i)/\mathcal{T})}{\sum_{i=1}^N \exp(\cos(V_{feat}^i, L_{feat}^i)/\mathcal{T})} \quad (4.1)$$

Additionally, the text to image loss is calculated using:

$$\mathcal{L}_{t2i} = \frac{1}{N} \sum_{i=1}^N -\log \frac{\exp(\cos(L_{feat}^i, V_{feat}^i)/\mathcal{T})}{\sum_{i=1}^N \exp(\cos(L_{feat}^i, V_{feat}^i)/\mathcal{T})} \quad (4.2)$$

where  $\cos(\cdot)$  is the cosine similarity function and  $L_{feat}^i$  denotes a feature embedding from the language modality of the  $i^{th}$  sample and  $V_{feat}^i$  is the same for a visual feature.

The symmetric InfoNCE loss is calculated as:

$$\mathcal{L}_{infoNCE} = \mathcal{L}_{i2t} + \mathcal{L}_{t2i} \quad (4.3)$$

With these changes, we expect our refined model architecture to capture the vehicle tracking features in a more efficient manner.

#### 4.3.4 SWIN Transformers

Swin Transformers [64] is a transformer-based deep learning model with state-of-the-art performance in vision tasks. Unlike the Vision Transformer (ViT) [21], Swin Transformer is highly efficient and has greater accuracy. ViT’s struggle with high resolution images since it’s computational complexity is quadratic to image size.

Using heirarchical feature maps, patch merging and a shifted window multi-headed self attention, SWIN transformers provide linear time complexity and perform well on high resolution images. We propose to use SWIN transformers replacing the visual encoder that takes the global motion image map as input.

### 4.4 EXPERIMENT SETUP

#### 4.4.1 Dataset

We use the CityFlow-NL dataset [25] to train and evaluate our model. It is the first multi-camera tracking dataset with natural language descriptions providing precise details for multi-view ground truth vehicle tracks. The CityFlow-NL dataset comprises approximately 4 hours of video surveillance. It consists data of 666 target vehicles collected from 40 cameras and a total of 3,028 single-view vehicle *tracks* and 5,289 unique natural language descriptions.

#### 4.4.2 Evaluation

For evaluation, we use the Mean Reciprocal Rank or MRR metric. For each input query, the model produces a ranked output list of *tracks*. The reciprocal rank of the  $i^{th}$  query is the multiplicative inverse of the rank of the first correct answer  $rank_i$ . MRR is the average of the reciprocal ranks of the overall test set

$$MRR = \frac{1}{N} \sum_{i=1}^N \frac{1}{rank_i} \quad (4.4)$$

For example, if the correct answer is ranked first, the reciprocal rank would be 1. If the correct answer is ranked third, the reciprocal rank would be 1/3. MRR is a useful metric because it takes into account both the accuracy of the ranking model (i.e., whether the correct answer is included in the list of predictions) and the position of the correct answer in the list of predictions. A high MRR indicates that the ranking model is both accurate and able to place the correct answers at the top of the list.

Model	MRR%
[115] (before post-processing)	40.73
Without BaiduNLP Augmentation	37.72
Backtranslation	38.20

Table 4.4: Comparison of Data Augmentation

Model	MRR%
[115] (before post-processing)	40.73
Symmetric Network [116]	43.92
Modified Symmetric Network	35.90
Modified Symmetric Network + Prompt Tuning	35.11
Modified Symmetric Network + SWIN	39.08
Modified Symmetric Network + Prompt Tuning + SWIN	39.45

Table 4.5: Ablation Study: Model MRR performance of different model architectures

### 4.4.3 Implementation Details

We build our code base upon [25]. For training, we utilize resources of 4 GPUs (G instances on AWS). During training, we use a batch size of 4 due to the limited computation power and train the model for 200 warm-up epochs with a learning rate of  $1e-5$  and for 500 epochs with a learning rate of  $1e-3$ . We use the AdamW optimizer [66], and use InfoNCE loss [95] as the criterion.

We use DistilBERT [101] as the language encoder. We use Visual encoders described in Sections 4.2.2 and 4.3.4. In order to extract video frames from videos, we use the data preprocessing technique provided by [116]. We use cropped car image files to generate IOU-filtered motion maps that include more vehicle information. The multimodal features from both encoders are fused by simply projecting them to the same space and averaging them.

## 4.5 RESULTS

Table 4.4 shows the variation in model performance with data augmentation. Although backtranslation does not perform better than BaiduNLP augmentation, we conclude that the language modality has a noticeable impact on vehicle retrieval. This was expected since the dataset only consisted of 3 natural language descriptions and augmenting it makes the model more robust and takes language more into consideration.

We perform an ablation study of our other experiments and show the results comparison in Table 4.5 and Figure 4.7. We notice that our novel model architecture performs comparably to existing SOTA methods.

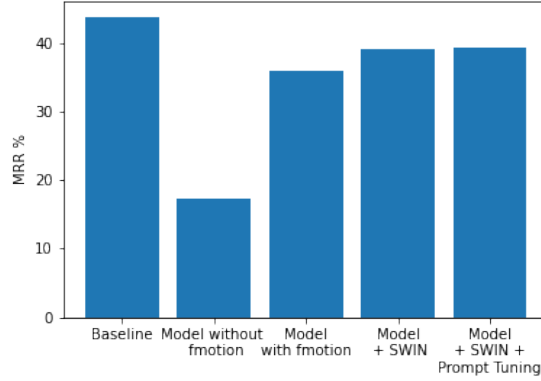


Figure 4.7: Model performance comparison

From Figure 4.7 we analyze the performance of our modified symmetric model with and without the feature  $f_{motion}$ . This feature is what is obtained by combining the feature vectors  $V_{motion}$  from the vision model and  $L_{motion}$  from the language encoder. Without these features, the model performance drops drastically to 17.41%. With the motion features, our model has an MRR of 35.9%. This was expected since simply capturing static features such as color, type will not provide information to the model about vehicle tracking coordinates. This shows the importance of dynamic properties and we argue that additional embeddings such as  $L_{speed}$  (to capture the speed of a vehicle) would further improve model performance. However, with the existing dataset there is not much information we can extract about such features from the language descriptions.

We also include the Prompt Tuning approach we introduced in Section 4.3.2 utilizing the manual template “A [COLOR] [TYPE] drives”. We append this feature to the projection head of our modified symmetric network. The intuition behind this approach was for feeding the model more explicit information about the color and type of the vehicle. However, we notice similar performance even with the addition of this feature. This further shows that static properties such as color and type do not contribute as greatly to the retrieval task. It consolidates our hypothesis that dynamic properties play a key role in the task.

By replacing existing visual encoders (EfficientNet B3) with the SWIN transformer we see a close to 3% increase in model performance. This essentially tells us that enhanced visual embeddings are able to obtain better static and dynamic properties.

Our ablation study further shows how adding prompt tuning to modified symmetric network + SWIN does not perform better. We believe that the SWIN transformer with its shifted window attention mechanism is able to better capture dynamic features especially better than existing methods. We propose that using SWIN in place of other visual transformers will in general lead to better performance on Vehicle Tracking tasks.

With this study we are able to answer the research questions we set out to address in Section 4.1. Our analysis shows how dynamic properties like motion, speed and position relative to other vehicles may lead to further improvements in the model. We also show that the language descriptions, although add noticeable features for tracking, does not contribute as much as visual features.

## 4.6 CONCLUSION

In this section, we build upon existing models for Vehicle tracking using natural language descriptions. We build our own symmetric model with modified visual encoders and introduce a motion feature from the language descriptions. Our model performs comparably to existing beatlines.

Furthermore, we perform an ablation study to show the contribution of the text and vision modalities to model performance. We also highlight the importance of dynamic properties for improved vehicle tracking compared to static properties such as color, vehicle type with experiments using manual prompt templates.

As a future research direction, we propose to explore methods in trying to capture dynamic features from the language description. We propose to use more sophisticated methods to capture  $L_{motion}$  and can also explore ideas to obtain information regarding vehicle velocity, position, etc. We can also explore an improvement of using Video-SWIN transformers [65] and using prompt learning instead of defining manual templates. Our prompt template can also include information of dynamic properties.

# CHAPTER 5

## TRAINING VISION-LANGUAGE TRANSFORMERS FROM CAPTIONS

### 5.1 INTRODUCTION

Vision-Language Transformers can be learned without low-level human labels (e.g. class labels, bounding boxes, etc). Existing work, whether explicitly utilizing bounding boxes [10, 89, 68] or patches [48], assumes that the visual backbone must first be trained on ImageNet [84] class prediction before being integrated into a multimodal linguistic pipeline. We show that this is not necessary and introduce a new model **Vision-Language from Captions (VLC)** built on top of Masked Auto-Encoders [37] that does not require this supervision. In fact, in a head-to-head comparison between ViLT, a strong patch-based vision-language transformer which is pretrained with supervised object classification, and our model, **VLC**, we find that our approach 1. outperforms ViLT on standard benchmarks, 2. provides more interpretable and intuitive patch visualizations, and 3. is competitive with many larger models that utilize ROIs trained on annotated bounding-boxes.

Our previous work has been evaluated by the industry and the link is [VLC](#). In previous work, we evaluate our model over several tasks including visual question answering, image-text retrieval and image classification. On top of this, we extend our work to provide more analysis and show our model can be applied to image-text grounding.

### 5.2 BACKGROUND

Image-text grounding refers to the process of associating textual descriptions with corresponding visual content. It plays a crucial role in multimodal learning and has significant importance in various real-world applications. Here are a few reasons why image-text grounding is important:

- **Enhancing Multimodal Understanding:** By linking textual and visual information, image-text grounding facilitates a deeper understanding of multimodal data. It enables the model to learn the relationships between different modalities, such as images and corresponding textual descriptions, leading to more comprehensive representations.
- **Visual Captioning:** Image-text grounding plays a vital role in generating descriptive captions for images. By grounding the textual descriptions to the visual content,

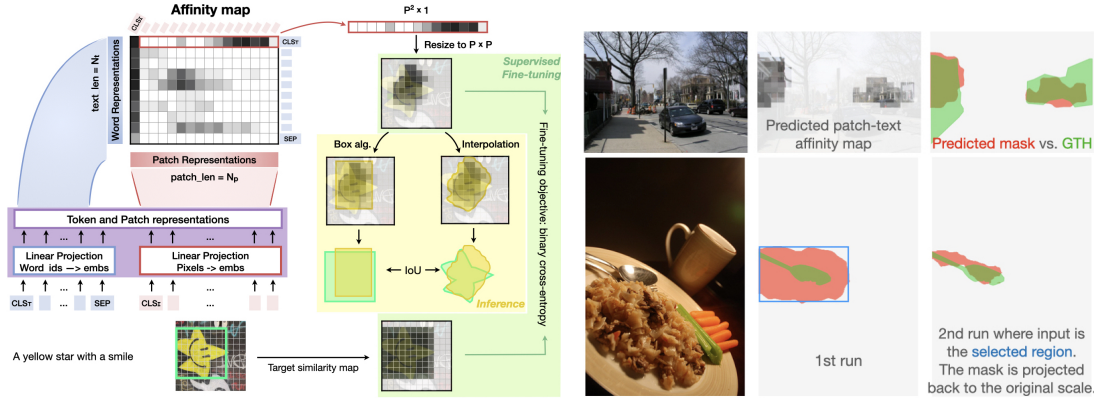


Figure 5.1: **Left:** Extending VLC to efficiently performing image-text grounding tasks: We finetune the Transformer backbone with supervision on similarities between patch and text representations. During inference, a bbox/mask can be obtained from the affinity scores with minimal computation overheads. **Right:** Reasonable segmentation masks resulted from interpolating patch-text affinities. Running the same inference procedure twice with coarse-to-fine resolution improves precision.

models can generate accurate and relevant captions that capture the salient details of the image. This has applications in accessibility, image understanding, and content summarization.

- **Content Understanding and Analysis:** In real-world applications like social media analysis, news aggregation, or brand monitoring, image-text grounding helps in understanding and analyzing the content shared by users. By grounding the text and images, it becomes possible to extract relevant information, detect sentiments, identify objects, and comprehend the context more accurately.

### 5.3 METHODOLOGY

Since the model was not pretrained to generate bounding boxes, we propose a novel algorithm which produces a bounding box based on patch-text affinities in a single forward pass. This results in much less inference time and higher parameter efficiency compared with competitive pipelines tailored to bounding box generation. See Figure 5.1 (Left) for an overview of the proposed grounding workflow.

#### 5.3.1 Finetuning

Our approach builds on the notion of an affinity map. Formally, we define the affinity map,  $\mathcal{A}$ , as the cosine distance between all  $\mathcal{L}$  language tokens and  $\mathcal{V}$  visual patches.

$$\hat{\mathcal{A}}_{t,p} = \cos(\mathcal{L}_t, \mathcal{V}_p) \quad (5.1)$$

for  $T$  tokens and  $P$  patches. This resulting matrix  $[-1, 1]^{T \times P}$  provides a normalized score for the relationship between every token and patch.



**Supervision Signals** For supervision, we translate bounding box annotations to sets of patches and phrases simply correspond to their indices, with [CLS] being the initial index 0. The ground-truth affinity scores can be obtained from labeled datasets where annotators selected bboxes corresponding to phrases in an (image, sentence) pair. Thus, for an (image, sentence) pair, we have a set of annotated (bbox, phrase) pairs, from which we would like to get a  $T \times P$  matrix of affinity scores. Note that the full image has dimensions  $H \times W$  and will be divided into a  $\sqrt{P} \times \sqrt{P}$  grid.

For a given annotated (bbox, phrase) pair, if the lexical token  $\mathcal{L}_t$  belongs to the phrase and patch  $\mathcal{V}_p$  overlaps with the bbox, we set  $\mathcal{A}_{t,p}$  in the ground-truth affinity map to be amount the patch overlaps with the bbox.

$$\mathcal{A}_{t,p} = \frac{|Patch \cap BBox|}{|Patch|} \quad (5.2)$$

If a bbox corresponds to the entire sentence, we set ground-truth affinity scores for the 0-th token, [CLS].

**Hyperparameters** We opt for a higher resolution (384) and smaller patches (16) since the patch size will determine the granularity of our predictions. This yields a final  $24 \times 24$  grid. We finetune VLC on the combined Refcoco+/g training sets for 50 epochs with AdamW [67] optimizer and a  $5e^{-4}$  learning rate.

**Finetuning Objective** We optimize our model with a binary-cross-entropy (BCE) loss between the ground-truth and predicted affinity scores.

$$Loss = \sum_{t \in T, p \in P} BCE(\mathcal{A}_{t,p}, \hat{\mathcal{A}}_{t,p}) \quad (5.3)$$

We intentionally avoided the use of a softmax because normalization would force competition across patches and affect their individual judgments. Each token-patch pair relationship is independent during loss calculation.

Note, that the domain expected by BCE is the full real line, but our affinities are computed via cosine similarities and are therefore bounded to  $[-1, 1]$ . Empirically, we found that re-scaling the input by  $k = 2$  improved performance due to a better utilization of the  $[0, 1]$  output range. However, performance decreases with a larger  $k$  because it tends to over-sharpen the sigmoid’s decision boundary, leaving less “wobble room” for the raw logits.

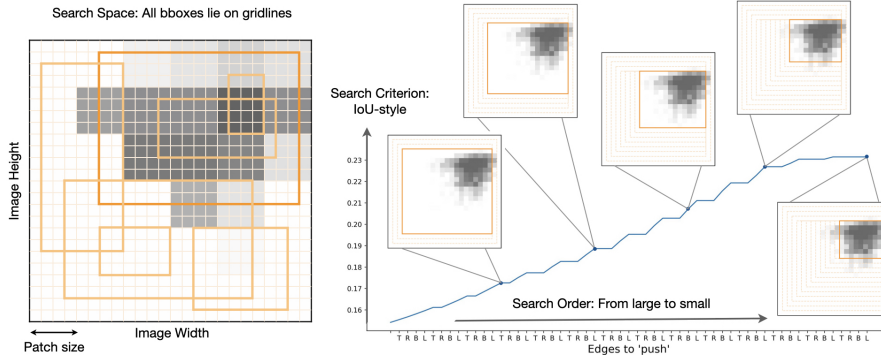


Figure 5.2: Inference procedure overview for “Affinity Map  $\Rightarrow$  BBox”, which is formulated as a search problem. Left: we define the search space to contain all candidate bboxes that lie on gridlines. Right: our algorithm greedily iterates over bboxes following the search order of large  $\rightarrow$  small, and terminates once the search criterion no longer improves.

### 5.3.2 Inference

The forward pass through the Transformer outputs a single tensor of shape  $L \times 768$ , or more accurately, the concatenation of two tensors of shape  $T \times 768$  and  $P \times 768$  for the encoded  $\mathcal{L}$ anguage and  $\mathcal{V}$ isual sequences, respectively. During inference, these matrices are used in Equation 5.1 to compute the predicted affinity matrix  $\hat{\mathcal{A}}$ . Next, we describe how a mask or a bounding box can be algorithmically derived from a predicted affinity matrix  $\hat{\mathcal{A}}$ . We emphasize that this procedure requires no additional learned parameters and is algorithmically efficient.

Since rows in  $\hat{\mathcal{A}}_{actv}$  associated with a phrase can be pooled into a single row and reshaped into  $\sqrt{P} \times \sqrt{P}$ , we henceforth refer  $\hat{\mathcal{A}}_{actv}^r$  to such a  $[-1, 1]^{\sqrt{P} \times \sqrt{P}}$  matrix with spatial correspondence to the image, albeit with a coarser resolution. Then, a binary mask can be simply derived by bilinearly interpolating  $\hat{\mathcal{A}}_{actv}^r$  back to the original resolution, and binarizing the values with a threshold.

The procedure for deriving a bbox from  $\hat{\mathcal{A}}_{actv}^r$  is more complicated, which we detail below. **We formulate bbox prediction as a search problem.** A bbox is denoted by  $\mathcal{B} = (x, y, w, h)$ , satisfying  $0 \leq x \leq x + w \leq W$ ,  $0 \leq y \leq y + h \leq H$ . The goal is to search for  $\mathcal{B}$  given  $\hat{\mathcal{A}}_{actv}^r$ , such that  $\mathcal{B}$  best represents a rectangular region  $\hat{\mathcal{A}}_{actv}^r$  tries to highlight (Figure 5.2, Left). To solve the search problem we need to decide on the following: the search space, the search order, and the search criterion.

**Search space** Theoretically, we have an infinite number of boxes to search over which can be as large as the input image or as small as nearly a single point. We constrain our search to only those boxes whose edges lie on gridlines, with step  $t$  denoting the unit between two gridlines.

**Search order** We start with a box encompassing the full image and greedily search for progressively smaller boxes.

**Search criterion** We propose a satisfying criterion  $\mathcal{M}$  that guarantees equivalence between  $\hat{\mathcal{A}}_{actv}^r$  and the theoretically optimal bbox ( $\mathcal{B}$ ). In the case where all values in  $\hat{\mathcal{A}}_{actv}^r$  are binary, this reduces to calculating a standard IoU (*Intersection-over-Union*). In practice, values will be real-valued. We overload the notation to handle the generalization from binary to real values in  $\hat{\mathcal{A}}_{actv}^r$ . Intersection is defined as the sum of the active values within  $\mathcal{B}$ . Union is equal to the sum of three terms, namely the active values inside  $\mathcal{B}$ , the values that should have been active inside  $\mathcal{B}$  and the active values outside  $\mathcal{B}$ . The first two terms in Union add up to  $|\mathcal{B}|$ . As a single equation this results in

$$\mathcal{M} = \frac{|\hat{\mathcal{A}}_{actv}^r \cap \mathcal{B}|}{|\mathcal{B}| + |\hat{\mathcal{A}}_{actv}^r \cap \neg\mathcal{B}|} \quad (5.4)$$

In practice, this is an easy calculation, as the intersections and negations simply correspond to summing the values inside and outside of the candidate box. Further, summing (in lieu of counting) trivially generalizes to the continuous valued activations.

We provide additional proof of the equivalence to optimizing F1 in the supplementary. We will note, there is a potential hyperparameter which we will call  $C$  that defines the desired “tightness” of the box. <sup>1</sup>

Next we leverage the the search space, order and criterion ( $\mathcal{M}$ ) in Algorithm 1 to predict bounding boxes. Initializing  $\mathcal{B}$  to surround the entire image, our *PUSH* algorithm greedily iterates over progressively smaller  $\mathcal{B}$  and terminates when  $\mathcal{M}(\hat{\mathcal{A}}_{actv}^r, \mathcal{B})$  no longer improves. Within each iteration, there are “push” attempts on the edges of  $\mathcal{B}$ , one at a time, by  $t$ -amount inwards according to an order  $\mathcal{O} \in Permutation([T, L, B, R])$ . For example,  $\mathcal{O} = [T, R, B, L]$  would translate to a clockwise progression of tightening edges from the *Top – Right – Bottom – Left*. A push operation is successful if and only if it increases  $\mathcal{M}$ . Otherwise, the edge is left unchanged. The algorithm terminates when it encounters unsuccessful push attempts on all four edges in a row. Figure 5.2 visualizes a *PUSH* execution.

In our experiments, we use values  $P = 24$ ,  $W = H = 384$  and  $t = \frac{1}{4} * patch\_size = 4$ .<sup>2</sup> A known limitation of our approach is highly discontinuous or concave affinity

<sup>1</sup>For example,  $C = 0.5 * |\hat{\mathcal{A}}_{actv}^r|$  would produce bboxes systematically smaller than  $C = |\hat{\mathcal{A}}_{actv}^r|$ . The supplementary discusses about where  $C$  is derived from. In this work, we leave the scaling to 1 and allow future work to analyze how  $C$  would influence the returned value of Algorithm.1

<sup>2</sup>Empirically, we find that the order of  $[L, B, R, T]$  does not affect performance. We ran our bbox inference procedure on the Refcog\_val(umd) 10 times, randomly permuting the order of  $[L, B, R, T]$  at

---

**Algorithm 1:** PUSH

---

**input** : heatmap  $\hat{\mathcal{A}}_{actv}^r$ , image size  $(W, H)$ , step  $t$ , measure  
 $\mathcal{M} : (\hat{\mathcal{A}}_{actv}^r, \mathcal{B}) \rightarrow \mathcal{R}$ ,  
order  $O \in \text{Permutation}([T, L, B, R])$   
**output** :  $\mathcal{B}$ : box coordinates  $(x, y, w, h)$  s.t.  $0 \leq x \leq x + w \leq W$  and  
 $0 \leq y \leq y + h \leq H$   
 $(x, y, w, h) \leftarrow (0, 0, W, H)$   
 $\mathcal{B} \leftarrow (x, y, w, h)$   
**while**  $w > 0$  and  $h > 0$  and  $move \neq \text{False}$  **do**  
    **for**  $e \in O$  **do**  
        **if**  $e == T$  **then**  $\mathcal{B}' \leftarrow (x, y + t, w, h - t)$   
        **else if**  $e == B$  **then**  $\mathcal{B}' \leftarrow (x, y, w, h - t)$   
        **else if**  $e == R$  **then**  $\mathcal{B}' \leftarrow (x, y, w - t, h)$   
        **else**  $\mathcal{B}' \leftarrow (x + t, y, w - t, h)$   
  
        **if**  $\mathcal{M}(\hat{\mathcal{A}}_{actv}^r, \mathcal{B}') > \mathcal{M}(\hat{\mathcal{A}}_{actv}^r, \mathcal{B})$  **then**  
             $\mathcal{B} \leftarrow \mathcal{B}'$   
             $(x, y, w, h) \leftarrow \mathcal{B}$   
             $move \leftarrow \text{True}$   
        **end**  
    **end**  
**end**  
**return**  $\mathcal{B}$ 

---

patterns. We did not observe this condition in our setting. We provide empirical evidence that our “Affinity Map  $\Rightarrow$  BBox” formalization is effective but leave a mathematically rigorous investigation to future work.

## 5.4 EXPERIMENTS

First we verify the strength of our approach in a zero-shot setting before investing in larger fine-tuning results. We adapt our model to perform zero-shot visual reasoning on the Kilogram [43] dataset. Kilogram challenges a model to recognize an abstract Tangram shape from a language description. Tangram shapes drastically differ from natural scenes as they only contain abstract and implicit visual clues. Hence, recognition demands more sophisticated reasoning about the interplay between visual and linguistic cues. The task is formulated into a 10-way classification problem where the model chooses the most relevant image given an abstract shape description. Specifically, we rank image relevance by the logits calculated by the pretrained image-text matching head. Table 5.1 shows that our model consistently outperforms ViLT across all input conditions. This supports our design consideration that freeing up the model from priors of limited ImageNet concepts helps it generalize to arbitrarily many linguistically-specified visual

---

each *PUSH* iteration. For 90% samples the algorithm returned the same predicted bbox all 10 times. For the remaining samples, the 10 returned bboxes have a joint IoU= $0.95 \pm 0.07$ .

Input Condition	ViLT	Ours	Human
WHOLE+BLACK	12.9	<b>13.8</b>	47.7
PARTS+BLACK	12.5	<b>15.2</b>	49.1
WHOLE+COLOR	11.7	<b>13.9</b>	49.5
PARTS+COLOR	10.7	<b>13.5</b>	63.0

Table 5.1: Zero-shot inference accuracy on Kilogram (dev), a challenging image-text matching task for recognizing abstract visual entities from linguistic descriptions. **VLC** consistently outperforms ViLT across all input conditions.

categories. Further improvement may come from additional finetuning to familiarize our model with the Tangram shape domain, which is left for future work.

We finetune the encoder on the training sets of Refcoco, Refcoco+, and Refcocog [109, 70] following the umd split. We convert ground-truth bounding boxes to patch-text affinity scores and use them to finetune the preceding representations. At inference time, we algorithmically predict a bounding box for each referential expression from affinity scores between the last layer patch and text representations.

**Comparison to modularized models** Previous models either use an RoI (Region-of-Interest) extractor to produce candidate bboxes to choose from [10, 26, 14, 87], or directly predict the coordinates and dimensions of bboxes in the image coordinate system [17]. The former approach relies on an off-the-shelf detector, while the latter requires gating mechanisms to infuse linguistic information into a visual backbone. Our approach outperforms all previous approaches with modular designs (Table ??, Top), meanwhile achieving the best inference-time efficiency. This justifies our unified approach in which both understanding and localization tasks can benefit from large-scale representation learning.

**Comparison to unified models that have far more parameters or computation** Most of performant models under comparison incorporated bounding box annotations from VG [49] in their pretraining data. With a much lighter architecture and much less annotated data, our model already achieves competitive performance. This holds promise that the performance will continue to improve as resolution and data are scaled up during pretraining.

**Generalization to dense prediction** Moreover, demonstrates that interpolating well-aligned path-text affinities naturally results in segmentation masks. Table 5.2 reports pixelwise-IoU between our interpolated masks and ground-truth masks on Refcoco+/g. We have doubled the performance of TSEG [86], the previous state-of-the-art method on Referring Expression Segmentation *without* training on ground-truth masks. Note,

Model	Refcoco+/g(umd) val		
TSEG-CRF [86]	25.95	22.62	23.41
<b>VLC</b> <sub>Base</sub> (ours)	50.65	45.88	46.37

Table 5.2: We achieve non-trivial pixelwise IoU on Referring Expression Segmentation through interpolating low-resolution affinity maps into segmentation masks without dense supervision.

performing a second forward pass with zoomed-in regions leads to significantly finer mask contours (Figure 5.1 Right). We leave a closer investigation of this inference trick to future work.

Model	Text	Image	Group
MTurk Human	89.50	88.50	85.50
Random Chance	25.00	25.00	16.67
UNITER <sub>Base</sub>	32.25	13.25	10.00
VILLA <sub>Base</sub>	30.00	12.00	8.00
ViLT <sub>Base</sub>	34.75	14.00	9.25
CLIP <sub>Base</sub>	30.75	10.50	8.00
FLAVA-ITM <sub>Base</sub>	32.25	<b>20.50</b>	14.25
<b>VLC</b> <sub>Base</sub>	28.00	19.75	12.50
UNITER <sub>Large</sub>	<b>38.00</b>	14.00	10.50
VILLA <sub>Large</sub>	37.00	13.25	11.00
<b>VLC</b> <sub>Large</sub>	32.00	20.00	<b>14.75</b>

Table 5.3: Winoground is a challenging test-only set for visio-linguistic compositional reasoning. **VLC**<sub>Large</sub> is competitive among similar-sized models without a second-stage pretraining.

**Better Grounding translates to compositional reasoning** We find that the finetuned patch-text affinities translate to greater compositional reasoning performance. We report inference-only results on Winoground [92] with our Refcoco-finetuned model in Table 5.3. Winoground requires pairing up two sets of images and sentences with minimally contrastive semantics. Instead of directly predicting a pairing logit using the image-text matching head, it is more effective to measure image-sentence association via grounding success conditioned on an input sentence. Concretely, by treating the max affinity as the image-sentence-matching score, our Refcoco-finetuned model achieves a state-of-the-art Group Score and outstanding Image Scores on Winoground.<sup>3</sup> This superior reasoning performance again verifies that our pretrained representations are more capable of using text/images to disambiguate each other.

<sup>3</sup>Please refer to the Winoground [92] paper for how their evaluation metrics are defined.











Input image	Predicted affinity map	Predicted reduced	bbox re-	Predicted & Ground-truth
				
<b>the leftmost train</b>	The model succeeded at distinguishing same-type instances via absolute position.			
				
<b>the smallest train</b>	The model succeeded at distinguishing same-type instances via absolute size.			

Table 5.4: Patch-token affinities can be finetuned towards localizing objects specified by referring expressions.

## 5.5 VISUALIZATIONS

We show that finetuning can further incentivize the affinity patterns to behave like bounding boxes, which is the standard output format for localization. We visualize predicted alignment after finetuning in Table. 5.4. We highlight a greater reasoning ability beyond recognition in terms of disambiguating visual entities based on how they are referenced linguistically.

## 5.6 CONCLUSION

We present **Vision-Language from Captions (VLC)**, a generic vision-language model pretrained with *only* image-caption pairs. It uses a single linear layer to project raw pixel stimuli and token embeddings into the same representation space, followed by Transformer blocks jointly modeling two modalities. By removing the dependency on image region proposals, our model is both (1) more data efficient, for it does not require pretraining-scale class labels or bounding box annotation, and (2) faster at inference, for it does not require a tedious vision-only branch.

Despite being lighter and faster, **VLC** performs competitively on a diverse set of vision-language tasks, as compared to existing approaches relying on detection or ImageNet supervision. The MIM pretraining objective encourages richer and language-aware visual representations, which implicitly results in finer-grained alignment between patch and token representations. With moderate downstream finetuning, **VLC** can be easily adapted to (1) perform multi-modal retrieval, (2) answer questions, (3) reason about visual information guided by free-form language, or (4) ground linguistically-referenced objects into bounding boxes. **VLC**'s strong performance across nine downstream benchmarks clearly demonstrate the wide task applicability of a unified vision-language encoder. As performance scales with increased training data, this opens an exciting avenue for

large-scale weakly-supervised open-domain vision-language models.



## APPENDIX I: GLOSSARY

### Chapter 2

1. **Activity Detection/Recognition:** The task of identifying and localizing activities of objects (person/vehicle) within an image or a video.
2. **Object Detection:** The task of identifying and localizing objects within an image or a video.
3. **Object Tracking:** The task of tracking and localizing target objects from a sequence of images or a video (containing multiple objects). A track refers to the camera id the vehicle is observed in, the frame number and the corresponding location coordinates (bounding boxes) of the object within the frame.
4. **Proposals of activity detection:** Clips of videos cropped from the raw footage.
5. **Backbone:** The main component of a neural network responsible for extracting low-level features from input data.
6. **Average Precision (AP):** A commonly used metric in object detection that measures the accuracy of object localization and classification.
7. **Intersection over Union (IoU):** A measurement of the overlap between the predicted bounding box and the ground truth bounding box, used to evaluate the accuracy of object localization.

### Chapter 3

1. **Streaming Perception:** The process of detecting and tracking objects in real-time video streams, particularly in the context of autonomous driving.
2. **Object Detection:** The task of identifying and localizing objects within an image or video.
3. **Receptive Field:** The region of an input image that influences the value of a particular pixel in the output feature map of a neural network.
4. **Feature Aggregation:** The process of combining multiple features from different layers or frames to create a more informative representation.
5. **Motion Consistency:** The ability to maintain the consistency of object motion across different frames in a video.

6. **Knowledge Distillation:** The process of transferring knowledge from a large, complex model (teacher) to a smaller, simpler model (student) to improve the student's performance.
7. **Inference:** The process of applying a trained model to new data to make predictions.
8. **Real-time Forecasting:** The ability to make predictions and decisions in real-time without significant delays or latency.
9. **Backbone:** The main component of a neural network responsible for extracting low-level features from input data.
10. **Neck:** An intermediate component in a neural network that connects the backbone and head, often used for feature fusion and refinement.
11. **Head:** The final component of a neural network responsible for generating predictions or outputs.
12. **Temporal Fusion:** The process of combining information from different frames in a video to capture temporal dependencies and correlations.
13. **Long-term Motion Consistency:** The ability to maintain consistency in object motion over a longer period of time, considering complex motion patterns and occlusions.
14. **Support Frame:** A previous frame used as reference or context for making predictions in a video stream.
15. **Fine-tuning:** The process of further training a pre-trained model on a specific task or dataset to improve its performance on that task.
16. **Average Precision (AP):** A commonly used metric in object detection that measures the accuracy of object localization and classification.
17. **Intersection over Union (IoU):** A measurement of the overlap between the predicted bounding box and the ground truth bounding box, used to evaluate the accuracy of object localization.

## Chapter 4

1. **Object Tracking:** The task of identifying and localizing objects within an image or a video. A track refers to the camera id the vehicle is observed in, the frame

number and corresponding location coordinates (bounding boxes) of the object within the frame.

2. **Backbone:** The main component of a neural network responsible for extracting low-level features from input data.
3. **Data augmentation:** Data augmentation is a technique used in machine learning and data science to increase the size and diversity of a dataset by creating new, synthetic data samples.
4. **Prompt tuning:** Prompt tuning refers to the process of fine-tuning or optimizing the initial prompt or instruction given to a language model to achieve desired outputs. It is commonly used with autoregressive language models, such as GPT (Generative Pre-trained Transformer), where the model generates text based on a given prompt.

## Chapter 5

1. **Embedding:** A vector to represent individual image or text description which is synonymous to feature/representation.
2. **Vision-language Pretraining (VLP):** The process to learn vision-language joint embeddings which can be applied to multimodal tasks.
3. **Modality:** A term referring to image/text/video data.
4. **Multi-modal:** The process of processing information from different modalities jointly.
5. **Visual question answering:** A task to answer questions based on images.
6. **Text-image grounding:** A task to ground objects in the images based on text descriptions.
7. **Generative models:** A group of models focus on modeling the joint distribution of inputs and outputs.
8. **Contrastive learning:** A Machine Learning paradigm where unlabeled data points are juxtaposed against each other to teach a model which points are similar and which are different.

## APPENDIX II: DATASETS

Dataset	Details	Reference
MEVA	Person and vehicle activity detection datasets.	[15]
VIRAT	Person and vehicle activity detection datasets.	[74]
ROAD	Vehicle activity detection datasets in ICCV 2021 ROAD Challenge.	[69]

Table 5: Datasets utilized in Chapter 2

Dataset	Details	Reference
Argoverse-HD	Urban outdoor scenes from two US cities	[52]
COCO	Object detection, segmentation, and captioning	[56]

Table 6: Datasets utilized in Chapter 3

Dataset	Details	Reference
CityFlow-NL	Vehicle tracking dataset	[25]
AI-CITY track 2 dataset	Vehicle tracking dataset in AI-CITY track 2	[72]

Table 7: Datasets utilized in Chapter 4

Dataset	Reference
MSCOCO	<a href="https://cocodataset.org/#home">https://cocodataset.org/#home</a>
VG	<a href="https://homes.cs.washington.edu/~ranjay/visualgenome/index.html">https://homes.cs.washington.edu/~ranjay/visualgenome/index.html</a>
GCC	<a href="https://ai.google.com/research/ConceptualCaptions/">https://ai.google.com/research/ConceptualCaptions/</a>
SBU	<a href="https://www.cs.rice.edu/~vo9/sbucaptions/">https://www.cs.rice.edu/~vo9/sbucaptions/</a>
VQA	<a href="https://visualqa.org/">https://visualqa.org/</a>
GQA	<a href="https://cs.stanford.edu/people/dorarad/gqa/about.html">https://cs.stanford.edu/people/dorarad/gqa/about.html</a>
Flickr30K	<a href="https://shannon.cs.illinois.edu/DenotationGraph/">https://shannon.cs.illinois.edu/DenotationGraph/</a>
OpenImages	<a href="https://storage.googleapis.com/openimages/web/index.html">https://storage.googleapis.com/openimages/web/index.html</a>
Kilogram	<a href="https://github.com/lil-lab/kilogram">https://github.com/lil-lab/kilogram</a>
Refcoco	<a href="https://github.com/lichengunc/refer">https://github.com/lichengunc/refer</a>
Winoground	<a href="https://huggingface.co/datasets/facebook/winoground">https://huggingface.co/datasets/facebook/winoground</a>

Table 8: Datasets utilized in Chapter 5

## REFERENCES

- [1] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Yvette Graham, Gareth J. F. Jones, , and Georges Quénot. Evaluating multiple video understanding and retrieval tasks at trecvid 2021. In *Proceedings of TRECVID 2021*. NIST, USA, 2021.
- [2] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quenot. TRECVID 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains. *arXiv:2104.13473 [cs]*, April 2021.
- [3] Shuai Bai, Zhedong Zheng, Xiaohan Wang, Junyang Lin, Zhu Zhang, Chang Zhou, Yi Yang, and Hongxia Yang. Connecting language and vision for natural language-based vehicle retrieval. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4029–4038, 2021.
- [4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1708–1718, 2021.
- [5] Hatem Belhassen, Heng Zhang, Virginie Fresse, and El-Bay Bourennane. Improving video object detection by seq-bbox matching. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, pages 226–233, 2019.
- [6] A. Bochkovskiy, C. Wang, and H. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv*, abs/2004.10934, 2020.
- [7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.
- [8] Xiaojun Chang, Wenhe Liu, Po-Yao Huang, Changlin Li, Fengda Zhu, Mingfei Han, Mingjie Li, Mengyuan Ma, Siyi Hu, and Guoliang Kang. MMVG-INF-Etrol@ TRECVID 2019: Activities in Extended Video. In *TREC Video Retrieval*

*Evaluation, TRECVID*, 2019.

- [9] Kai Chen, Jiaqi Wang, Shuo Yang, Xingcheng Zhang, Yuanjun Xiong, Chen Change Loy, and Dahua Lin. Optimizing video object detection via a scale-time lattice. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7814–7823, 2018.
- [10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.
- [11] Zhi-Qi Cheng, Qi Dai, Siyao Li, Teruko Mitamura, and Alexander Hauptmann. Gsrformer: Grounded situation recognition transformer with alternate semantic attention refinement. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, pages 3272–3281, 2022.
- [12] Zhi-Qi Cheng, Xiao Wu, Siyu Huang, Jun-Xiu Li, Alexander G Hauptmann, and Qiang Peng. Learning to transfer: Generalizable attribute learning with multitask neural model search. In *Proceedings of the ACM international conference on Multimedia (ACM MM)*, pages 90–98, 2018.
- [13] Ting-Wu Chin, Ruizhou Ding, and Diana Marculescu. Adascale: Towards real-time video object detection using adaptive scaling. In *Proceedings of Machine Learning and Systems (MLSys)*, 2019.
- [14] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021.
- [15] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1059–1067, Waikoloa, HI, USA, January 2021. IEEE.
- [16] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.
- [17] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1769–1779, 2021.

- [18] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks. In *International Conference on Computer Vision (ICCV)*, pages 1911–1920, 2019.
- [19] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10886–10895, 2021.
- [20] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13733–13742, 2021.
- [21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.
- [22] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.
- [23] Yunhao Du, Binyu Zhang, Xiang Ruan, Fei Su, Zhicheng Zhao, and Hong Chen. Omg: Observe multiple granularities for natural language-based vehicle retrieval. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3123–3132, 2022.
- [24] Christoph Feichtenhofer. X3D: Expanding Architectures for Efficient Video Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–210, Seattle, WA, USA, June 2020. IEEE.
- [25] Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. Cityflow-nl: Tracking and retrieval of vehicles at city scale by natural language descriptions. *ArXiv*, abs/2101.04741, 2021.
- [26] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616–6628, 2020.
- [27] Kaifeng Gao, Long Chen, Yifeng Huang, and Jun Xiao. Video relation detection

- via tracklet based visual transformer. In *Proceedings of ACM Conference on Multimedia (ACM MM)*, pages 4833–4837. ACM, 2021.
- [28] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. OTA: optimal transport assignment for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 303–312, 2021.
- [29] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021.
- [30] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12038–12047, Long Beach, CA, USA, June 2019. IEEE.
- [31] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. NAS-FPN: learning scalable feature pyramid architecture for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7036–7045, 2019.
- [32] A. Ghosh, A. Nambi, A. Singh, and et al. Adaptive streaming perception using deep reinforcement learning. *CoRR*, abs/2106.05665, 2021.
- [33] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S. Huang. Seq-nms for video object detection. *CoRR*, abs/1602.08465, 2016.
- [34] Jun-Yan He, Zhi-Qi Cheng, Chenyang Li, Wangmeng Xiang, Binghui Chen, Bin Luo, Yifeng Geng, and Xuansong Xie. Damo-streamnet: Optimizing streaming perception in autonomous driving. *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)*, 2023.
- [35] Jun-Yan He, Shi-Hua Liang, Xiao Wu, Bo Zhao, and Lei Zhang. Mgseg: Multiple granularity-based real-time semantic segmentation network. *IEEE Transactions on Image Processing (TIP)*, 30:7200–7214, 2021.
- [36] Jun-Yan He, Xiao Wu, Zhi-Qi Cheng, Zhaoquan Yuan, and Yu-Gang Jiang. Db-lstm: Densely-connected bi-directional lstm for human action recognition. *Neurocomputing*, 444:319–331, 2021.
- [37] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *CVPR*, 2022.



- [38] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, February 2020.
- [39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE.
- [40] Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing. In *Conference on Empirical Methods in Natural Language Processing*, 2015.
- [41] Siyu Huang, Xi Li, Zhi-Qi Cheng, Zhongfei Zhang, and Alexander Hauptmann. Gnas: A greedy neural architecture search method for multi-attribute learning. In *Proceedings of the 26th ACM international conference on Multimedia (ACM MM)*, pages 2049–2057, 2018.
- [42] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, and Marcelo H. Ang Jr. Tada! temporally-adaptive convolutions for video understanding. In *Proceedings of International Conference on Learning Representations, (ICLR)*, 2022.
- [43] Anya Ji, Noriyuki Kojima, Noah Rush, Alane Suhr, Wai Keen Vong, Robert D Hawkins, and Yoav Artzi. Abstract visual reasoning with tangram shapes. *arXiv preprint arXiv:2211.16492*, 2022.
- [44] Yiqi Jiang, Zhiyu Tan, Junyan Wang, Xiuyu Sun, Ming Lin, and Hao Li. Giraffedet: A heavy-neck paradigm for object detection. In *International Conference on Learning Representations (ICLR)*, 2022.
- [45] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 817–825, 2016.
- [46] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset. *arXiv:1705.06950 [cs]*, May 2017.
- [47] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction

- for object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *European Conference on Computer Vision (ECCV)*, volume 12370, pages 355–371, 2020.
- [48] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.
- [49] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.
- [50] Jin-Peng Lan, Zhi-Qi Cheng, Jun-Yan He, Chenyang Li, Bin Luo, Xu Bao, Wangmeng Xiang, Yifeng Geng, and Xuansong Xie. Procontext: Exploring progressive context transformer for tracking. *arXiv preprint arXiv:2210.15511*, 2022.
- [51] Chenyang Li, Zhi-Qi Cheng, Jun-Yan He, Pengyu Li, Bin Luo, Hanyuan Chen, Yifeng Geng, Jin-Peng Lan, and Xuansong Xie. Longshortnet: Exploring temporal and semantic features fusion in streaming perception. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.
- [52] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12347, pages 473–488, 2020.
- [53] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2657–2664, 2014.
- [54] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal Shift Module for Efficient Video Understanding. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7082–7092, Seoul, Korea (South), October 2019. IEEE.
- [55] Lijian Lin, Haosheng Chen, Honglun Zhang, Jun Liang, Yu Li, Ying Shan, and Hanzi Wang. Dual semantic fusion network for video object detection. In *ACM International Conference on Multimedia (ACM MM)*, pages 1855–1863, 2020.
- [56] T. Lin, M. Maire, S. Belongie, and et al. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, volume 8693, pages 740–755, 2014.

- [57] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944. IEEE Computer Society, 2017.
- [58] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [59] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys (CSUR)*, 2021.
- [60] S. Liu, L. Qi, H. Qin, and et al. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018.
- [61] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8759–8768. Computer Vision Foundation / IEEE Computer Society, 2018.
- [62] Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Lijun Yu, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, Peng Chen, and Alexander G. Hauptmann. Argus: Efficient Activity Detection System for Extended Video Analysis. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 126–133, Snowmass Village, CO, USA, March 2020. IEEE.
- [63] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.
- [64] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.
- [65] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3192–3201, 2021.
- [66] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In

*International Conference on Learning Representations*, 2017.

- [67] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018.
- [68] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 2019.
- [69] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.
- [70] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.
- [71] Niluthpol Chowdhury Mithun, Juncheng Billy Li, Florian Metze, and Amit K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, 2018.
- [72] Milind R. Naphade, Shuo Wang, D. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Archana Venkatachalapathy, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Alice Li, Sha Li, and Ramalingam Chellappa. The 6th ai city challenge. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3346–3355, 2022.
- [73] Tien-Phat Nguyen, Ba-Thinh Tran-Le, Xuan-Dang Thai, Tam V. Nguyen, Minh N. Do, and Minh-Triet Tran. Traffic video event retrieval via text query using vehicle appearance and motion attributes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4160–4167, 2021.
- [74] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*,

pages 3153–3160, June 2011. ISSN: 1063-6919.

- [75] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018.
- [76] Mengshi Qi, Jie Qin, Yi Yang, Yunhong Wang, and Jiebo Luo. Semantics-aware spatial-temporal binaries for cross-modal video retrieval. *IEEE Transactions on Image Processing*, 30:2989–3004, 2021.
- [77] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. ELECTRICITY: An Efficient Multi-camera Vehicle Tracking System for Intelligent City. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2511–2519, Seattle, WA, USA, June 2020. IEEE.
- [78] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Trm: Temporal relocation module for video recognition. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, 2022.
- [79] Yijun Qian, Lijun Yu, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Adaptive Feature Aggregation for Video Object Detection. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 143–147, Snowmass Village, CO, USA, March 2020. IEEE.
- [80] Jian-Jun Qiao, Zhi-Qi Cheng, Xiao Wu, Wei Li, and Ji Zhang. Real-time semantic segmentation with parallel multiple views feature augmentation. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, pages 6300–6308, 2022.
- [81] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
- [82] Hamid Rezaatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, 2019.
- [83] Mamshad Nayeem Rizve, Ugur Demir, Praveen Tirupattur, Aayush Jung Rana, Kevin Duarte, Ishan R Dave, Yogesh S Rawat, and Mubarak Shah. Gabriella: An Online System for Real-Time Activity Detection in Untrimmed Security Videos. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages

4237–4244, January 2021. ISSN: 1051-4651.

- [84] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.
- [85] Gurkirt Singh, Stephen Akrigg, Manuele Di Maio, Valentina Fontana, Reza Javanmard Alitappeh, Suman Saha, Kossar Jeddisaravi, Farzad Yousefi, Jacob Culley, Tom Nicholson, Jordan Omokeowa, Salman Khan, Stanislao Grazioso, Andrew Bradley, Giuseppe Di Gironimo, and Fabio Cuzzolin. ROAD: The ROad event Awareness Dataset for Autonomous Driving. *arXiv:2102.11585 [cs]*, February 2021.
- [86] Robin Strudel, Ivan Laptev, and Cordelia Schmid. Weakly-supervised segmentation of referring expressions. *arXiv preprint arXiv:2205.04725*, 2022.
- [87] Sanjay Subramanian, William Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5198–5215, 2022.
- [88] Guanxiong Sun, Yang Hua, Guosheng Hu, and Neil Robertson. MAMBA: multi-level aggregation via memory bank for video object detection. In *Proceedings of AAAI Conference on Artificial Intelligence, (AAAI)*, pages 2620–2627, 2021.
- [89] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *EMNLP*, 2019.
- [90] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019.
- [91] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787, 2020.
- [92] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022.
- [93] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar

- Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, Salt Lake City, UT, June 2018. IEEE.
- [94] Shuyuan Tu, Qi Dai, Zuxuan Wu, Zhi-Qi Cheng, Han Hu, and Yu-Gang Jiang. Implicit temporal modeling with learnable alignment for video recognition. *arXiv preprint arXiv:2304.10465*, 2023.
- [95] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.
- [96] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An improved one millisecond mobile backbone. *CoRR*, abs/2206.04040, 2022.
- [97] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *CoRR*, abs/2207.02696, 2022.
- [98] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 43(10):3349–3364, 2021.
- [99] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 20–36, Cham, 2016. Springer International Publishing.
- [100] Shiyao Wang, Yucong Zhou, Junjie Yan, and Zhidong Deng. Fully motion-aware network for video object detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proceedings of European Conference on Computer Vision (ECCV)*, volume 11217, pages 557–573, 2018.
- [101] Zeyu Wang, Yu Wu, Karthik Narasimhan, and Olga Russakovsky. Multi-query video retrieval. *ArXiv*, abs/2201.03639, 2022.
- [102] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards Real-Time Multi-Object Tracking. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12356, pages 107–122. Springer International Publishing, Cham, 2020. Series

Title: Lecture Notes in Computer Science.

- [103] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.
- [104] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [105] Fanyi Xiao and Yong Jae Lee. Video object detection with an aligned spatial-temporal memory. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume 11212, pages 494–510, 2018.
- [106] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin P. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*, 2017.
- [107] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin P. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.
- [108] Jinrong Yang, Songtao Liu, Zeming Li, Xiaoping Li, and Jian Sun. Real-time object detection for streaming perception. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5385–5395, 2022.
- [109] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016.
- [110] Lijun Yu, Peng Chen, Wenhe Liu, Guoliang Kang, and Alexander G. Hauptmann. Training-free Monocular 3D Event Detection System for Traffic Surveillance. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3838–3843, December 2019.
- [111] Lijun Yu, Qianyu Feng, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. Zero-VIRUS<sup>\*</sup> : Zero-shot Vehicle Route Understanding System for Intelligent Transportation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2534–2543, Seattle, WA, USA, June 2020. IEEE.
- [112] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann. CMU Informedia



- at TRECVID 2020: Activity Detection with Dense Spatio-temporal Proposals. In *TREC Video Retrieval Evaluation, TRECVID*, page 9, 2020.
- [113] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G Hauptmann. CMU Informedia at TRECVID 2021: Activity Detection with Argus++. In *TREC Video Retrieval Evaluation, TRECVID*, 2021.
- [114] Lijun Yu, Dawei Zhang, Xiangqun Chen, and Alexander Hauptmann. Traffic Danger Recognition With Surveillance Cameras Without Training Data. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, November 2018.
- [115] Jiacheng Zhang, Xiangru Lin, Minyue Jiang, Yue Yu, Chenting Gong, Wei Zhang, Xiao Tan, Yingying Li, Errui Ding, and Guanbin Li. A multi-granularity retrieval system for natural language-based vehicle retrieval. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3215–3224, 2022.
- [116] Chuyang Zhao, Haobo Chen, Wenyuan Zhang, Junru Chen, Sipeng Zhang, Yadong Li, and Boxun Li. Symmetric network with spatial relationship modeling for natural language-based vehicle retrieval. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3225–3232, 2022.
- [117] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-Guided Feature Aggregation for Video Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 408–417, Venice, October 2017. IEEE.
- [118] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of IEEE Conference on Computer Vision (ICCV)*, pages 408–417, 2017.
- [119] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4141–4150, 2017.