Watch Only Once: An End-to-End Video Action Detection Framework

Shoufa Chen[1] Peize Sun[1] Enze Xie[1] Chongjian Ge[1] Jiannan Wu[1]
Lan Ma[2,3] Jiajun Shen[2,3] Ping Luo[1]

[1] The University of Hong Kong [2] TCL Research
[3] HKU-TCL Joint Research Centre for Artificial Intelligence

Abstract

*We propose an end-to-end pipeline, named **W**atch **O**nce **O**nly (WOO), for video action detection. Current methods either decouple video action detection task into separated stages of actor localization and action classification or train two separated models within one stage. In contrast, our approach solves the actor localization and action classification simultaneously in a unified network. The whole pipeline is significantly simplified by unifying the backbone network and eliminating many hand-crafted components. WOO takes a unified video backbone to simultaneously extract features for actor location and action classification. In addition, we introduce spatial-temporal action embeddings into our framework and design a spatial-temporal fusion module to obtain more discriminative features with richer information, which further boosts the action classification performance. Extensive experiments on AVA and JHMDB datasets show that WOO achieves state-of-the-art performance, while still reduces up to 16.7% GFLOPs compared with existing methods. We hope our work can inspire re-thinking the convention of action detection and serve as a solid baseline for end-to-end action detection. Code is available at* https://github.com/ShoufaChen/WOO.

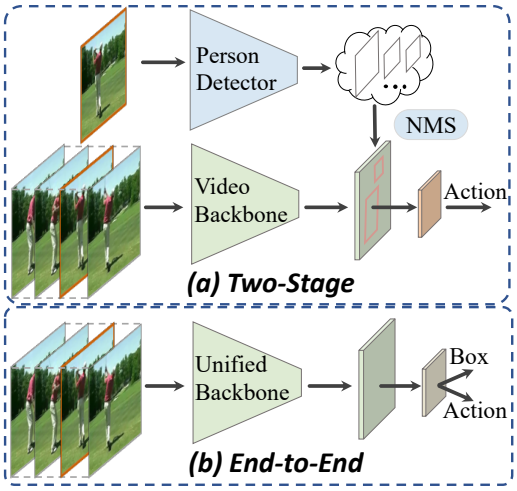

Figure 1: **Motivation of WOO. (a)** Previous dominant video action detection methods usually adopt two separate networks: an independent 2D detection model for actor localization from every key frames, and a 3D video model for action classification from video clips. **(b)** Our end-to-end unified framework uses a single backbone network to handle both 2D image detection and 3D video classification (*i.e.* 2D spatial dimensions plus a temporal dimension). This unified backbone only "watches" an input video once, and directly produces both actor localization and action classification.

1. Introduction

Video action detection consists of actor bounding box localization and action type classification. It has a significant impact on applications such as robotics, security, health, and so on. Although considerable progress on accuracy performance has been made, the complex and isolated pipelines of existing methods obstruct their scalability and practicality in the real world.

The complexity of current methods comes from a fundamental dilemma between actor localization and action classification, that is, *using a single key frame is "positive" for actor localization but "negative" for action classification, while using multiple frames has a reverse impact.* This is because actor localization requires a 2D detection model to predict actor bounding boxes on the key frame of a video clip. At this stage, taking neighboring frames of the clip into account brings extra computation and memory cost as well as localization noise. In contrast, action classification heavily relies on a 3D video model to extract temporal knowledge embedded in the video sequence. A single key frame brings a poor temporal motion representation for action classification.

Two possible workarounds are proposed by previous

methods to relieve this dilemma. The first one [7, 8, 38, 37, 25] uses an off-the-shelf person detector, which is not jointly trained with action classification models, to generate actor proposals. Then, an independent video model adopts these actor proposals and the raw frames as input to predict action classes. The person detector model alone is complicated enough, which is pre-trained on the ImageNet [4] and COCO [22] human keypoint detection, and further fine-tuned on the target action detection dataset. This workaround has a complex and computational-expensive pipeline, which requires two separate models and two training stages. Furthermore, the separate optimization on two sub-problems leads to a sub-optimal solution [19].

The second type of methods [30, 19] jointly train the actor detection and action classification models in a single training stage. Although the training pipeline is simplified to some extent, both of these two models still need to extract features independently and directly from raw images. Thus, the overall framework still brings a heavy computation and memory cost.

A natural question is: *"Is it possible to design a simple and unified framework to simultaneously address actor localization and action classification in a single end-to-end model?"*

This paper answers this question by proposing a novel video framework, termed ***W**atch **O**nce **O**nly* (***WOO***), which solves video action detection in an end-to-end manner. WOO directly predicts coordinates of actor's bounding boxes as well as probabilities of action classes from a video clip, as illustrated in Figure 1(b). Benefiting from our simple design, one could "watch" a video clip only once and predict where the actors are and what they are doing. Specifically, our method consists of three key components, including a unified backbone, a spatial-temporal action embedding, and a spatial-temporal knowledge fusion mechanism.

First, we design a simple yet effective module that enables a single backbone network to provide task-specific feature maps for both the actor localization head and the action classification head. This module is light-weight and used to isolate the key frame features from the features of all frames in the early stage of the backbone network. The motivation is that the key frame feature gets more interaction with the neighboring frames as the model going deeper. The proposed module can be easily plugged into many existing video backbones, such as single-pathway I3D [18, 3], SlowOnly [8], X3D [7], and two-pathway SlowFast [8].

Second, we notice that the unified architecture tends to behave well for actor localization, but is still limited for action classification. As observed in [12, 8], the difficulty in action detection mainly lies in action classification. Thus, we suspect that a single backbone for both tasks would bias towards localization and find an undesired solution, thus hurting the performance of action classification. Based on this observation, we propose spatial and temporal action embedding and the interaction mechanism between them, to make the action classification features more discriminative in both spatial and temporal perspectives.

Third, we further propose a spatial-temporal fusion module to aggregate spatial and temporal knowledge together. The spatial properties such as shape and pose, as well as the temporal properties such as dynamic motion and the temporal scale of action are combined through our spatial-temporal fusion module, to generate the action features for action classification.

Our main contributions are as follows.

1. We propose an end-to-end framework for video action detection, which directly produces the bounding boxes and action classes simultaneously, given a video clip as input. Our framework does not need an independent person detector, which is indispensable in existing works [7, 8, 38, 37, 25].

2. We propose a spatial-and-temporal embedding, and an embedding interaction mechanism, which improve discriminativeness of the features for action classification. A spatial-temporal fusion module is further proposed to aggregate features from spatial and temporal dimensions.

3. Extensive experiments on AVA and JHMDB demonstrate that the performance of WOO could outperform or on par with previous well-established and complicated two-stage action detectors, while still reducing up to 16.7% FLOPs.

2. Related Work

Two-Stage, Two-Backbone. Current state-of-the-art models for video action detection usually adopt a two-stage pipeline with two backbones [7, 8, 38, 37, 25]. These methods simply split video action detection task into actor localization and action classification. More specifically, in the first stage, they pre-train a model on COCO keypoint [22] and then fine-tune it on the target video action detection dataset. In the second stage, they take the key frame of a video clip as the input of the detection model obtained in the first stage to predict actor bounding boxes. Then they take a video clip and the actor bounding boxes as the input of the 3D video backbone network to extract features in the region of interest (RoI) for action class prediction. Naturally, these methods suffer from heavy complexity and low efficiency due to the sequential training stage and separate model architectures. Moreover, The independent optimization on the two independent stages may lead to a sub-optimal solution.
One-Stage, Two-Backbone. YOWO [19] and ACRN [30] simplified the pipeline by training the 2D actor detection

model and 3D video model simultaneously. However, there are still two separate models to optimize. Taking YOWO [19] for example, it contains a 3D model pre-trained on Kinetics [18] and a 2D model, YOLO [27] pre-trained on PASCAL VOC [5]. The heavy computation and memory burden still exist, although the pipeline is simplified to some extent.

In contrast to these methods, we propose an end-to-end video action detection framework, called WOO. Our WOO is refreshingly simple: It directly predicts actor bounding boxes and the corresponding action classes given a video clip. Only a single union backbone is adopted in our framework. Thus, in our method, the video clip only needs to be watched once only.

End-to-end object detection. Recently, some end-to-end object detection frameworks [2, 42, 31] are proposed to directly output the predictions without any hand-crafted label assignment or post-processing procedure, such as non-maximum suppression (NMS), achieving fantastic performance. Among these works, DETR [2] can be viewed as the first end-to-end object detection method which adopts the global attention mechanism [33] and the bipartite matching between predictions and ground truth objects. While DETR discards the NMS procedure and achieves remarkable overall performance, it suffers from much a longer training duration than mainstream detectors and lower performance on the small objects. To solve above mentioned issues in DETR, Deformable-DETR [42] is proposed to restrict each object query to a small set of crucial sampling points around the reference points, instead of all points in the feature map. Deformable-DETR is efficient and fast-converging. Concurrent with [42], Sparse R-CNN [31] utilizes a fixed sparse set of learned object proposals and performs classification and localization in an iterative way. Sparse R-CNN demonstrates accuracy, run-time and training convergence performance on par with the well-established detectors. In this work, we directly adopt the detection head of Sparse R-CNN [31] for localization.

Attention mechanism for action recognition. First proposed for language related tasks [33, 40], the attention mechanism has been a fairly popular concept and a helpful tool in both the natural language processing (NLP) and computer vision (CV) community in recent years. As for action recognition, Non-local Networks [35] leverage self-attention to capture dependencies between features at different time or space, making attention mechanism applicable for action classification. [37] leverages non-local block as a long-term feature bank operator, which enables video models with access to long-term information and improves action detection performance.

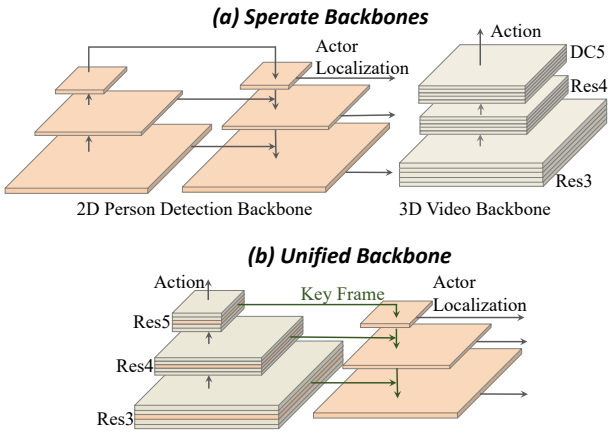

Figure 2: **Comparisons of backbone architecture.** ***(a)*** Two separate backbones for actor localization and action classification. Video backbone adopts res5 stage with dilated convolution (DC5). ***(b)*** A single union backbone which can provide task-specific features for actor localization and action classification simultaneously, enabling nearly cost-free feature extraction for actor localization compared to ***(a)***. Key frame features are illustrated in light orange color. Here we purposely omit the res2 features for visual simplicity.

3. Method

3.1. Overview

In contrast to previous work, we adopt an end-to-end method to solve the video action detection problem, both in training and evaluation. Without any post processing (e.g., non-maximum suppression (NMS)), our proposed model is able to directly output actor bounding boxes and action classes given a video clip.

Formally, let $X \in \mathbb{R}^{C\times T\times H\times W}$ denote the input spatial-temporal feature map for a specific layer, where C stands for number of channels, T for time, H and W for spatial height and width, respectively. Following the previous works [7, 8, 37], we place the key frame in the middle of a video clip in this work. We will use $X_{t=\lfloor T/2 \rfloor} \in \mathbb{R}^{C\times H\times W}$ to indicate the key frame feature map for further discussion.

3.2. Union Backbone

We aim to propose a simple and unified video backbone that provides task-specific features for both localization and classification. In previous video backbones [8, 38, 7, 37], the key frame features would interact with neighboring frame features by temporal pooling or 3D convolution with temporal kernel size larger than one, which would add undesired disturbance to the key frame features. To overcome this problem, our video backbone is designed to isolate the

key frame features from the early stage of the network before temporal interaction works. Figure 2 ***(b)*** illustrates our proposed simple yet effective backbone structure.

As opposed to the previously widely used backbone, SlowFast [8], which sets the spatial stride of res5 to 1 and uses a dilation of 2 for its filters to increase the spatial resolution of res5 by 2× (see Figure 2 ***(a)***), we remove the dilated convolution in res5 and adopt the Feature Pyramid Network [21] structure for key frame feature extraction. The FPN module adopts the key frame feature output by res2, res3, res4, res5 as the input. We further use the output features of the FPN module for actor localization and the features output by res5 for action classification. To this end, an unified action backbone is established to provide task-relevant features.

There are several benefits of the above design pattern. Firstly, the actor localization head adopts the hierarchical feature representations as source features, which are quite advantageous for object detection [21, 34, 32, 1, 23].

Secondly, the key frame features used for actor localization are isolated from features of all video frames through the FPN structure, starting at the early stage (*i.e.*, res2) of the backbone. This implementation can reduce the interference from neighboring frames because the key frame feature gets more interaction with the neighboring frames as the model goes deeper.

Thirdly, compared to existing two-backbone methods [7, 8, 37] that use an totally independent convolutional network (*e.g.* Faster R-CNN [28] with a ResNeXt-101-FPN [39, 21] backbone) for actor localization, we only add a light-weight FPN module that tasks image features as input, which reduces the parameters and FLOPs remarkably.

Furthermore, the above design is independent of the video backbone architectures and the module can be easily plugged into various video backbones [8, 7].

3.3. Actor Localization Head

Inspired by the recent advanced Sparse R-CNN [31], we design an end-to-end actor detection head for actor localization. Receiving the hierarchical features generated by the FPN module in Figure 2(b), the detection head is able to predict the bounding box coordinates and corresponding scores indicating the model's confidence on the box containing an actor.

Moreover, the person detector utilizes set prediction loss for optimal bipartite matching between prediction and ground truth at the training stage and does not need post-process (*e.g.* NMS) at the evaluation stage. Furthermore, unlike the two-backbone methods, we do not need extra ImageNet [4] or COCO [22] pre-training because the person detector shares a backbone with the action classifier.

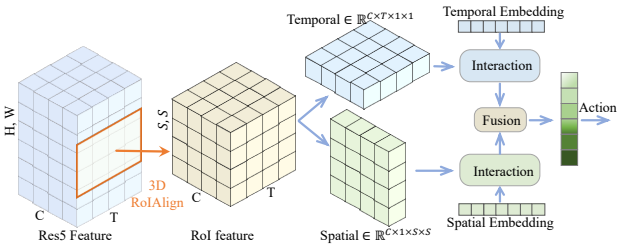

Figure 3: **Action classification head.** Given the RoI feature of a specific box for T frames, spatial and temporal action features are generated. Then, spatial and temporal embedding is used to make action feature representation more discriminative through the interaction module. Finally, the multi-layer perceptron (MLP) takes as input the fused spatial-temporal feature and predicts the action class logits. See text for details.

3.4. Action Classification Head

Given N actor proposal boxes generated by the aforementioned person detector, we use RoIAlign [13] to extract *spatial* and *temporal* features for each box. Then these two types of features are fused together to generate the final action class prediction. We will detail this process as follows.

Spatial Action Features. Let $X_5 \in \mathbb{R}^{C\times T\times H\times W}$ denote the output feature volume generated by res5. We perform global average pooling along the temporal dimension to obtain a spatial feature map $f^s \in \mathbb{R}^{C\times 1\times H\times W}$. The RoIAlign is applied on f^s with N actor proposal boxes, producing N spatial RoI features, *i.e.*, $f_1^s, f_2^s, \cdots, f_N^s \in \mathbb{R}^{C\times S\times S}$, where $S \times S$ is the spatial output size of RoIAlign.

Temporal Action Features. In addition to spatial action features, temporal properties are critical, especially in video data. To capture the temporal motion information, we extract temporal features from every frame in the feature volume X_5. Since we mainly focus on the temporal information here, we employ a global average pooling on the spatial dimension to efficiently extract the temporal RoI features. Formally, the temporal action feature is denoted as $f_1^t, f_2^t, \cdots, f_N^t \in \mathbb{R}^{C\times T\times 1\times 1}$.

Embedding Interaction. To obtain more discriminative features, we further introduce spatial and temporal embedding to be convolved with the aforementioned spatial and temporal features for the purpose of enriching instance characteristics. The spatial embedding is expected to encode the spatial properties such as shape, pose, etc. The temporal embedding is to encode the temporal dynamic properties such as dynamics and the temporal scale of an action. Note that the embedding is exclusive for each of the N features. Thus, we define $E^s \in \mathbb{R}^{N\times d}, E^t \in \mathbb{R}^{N\times d}$ for spatial and temporal embedding. The $E_n^s \in \mathbb{R}^d, E_n^t \in \mathbb{R}^d$

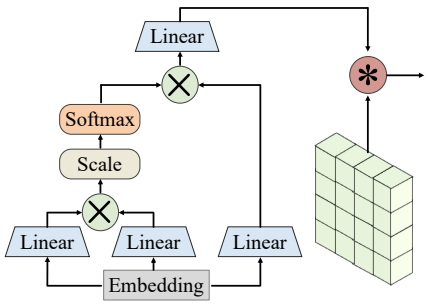

Figure 4: **Structure of interaction module.** Here we plot spatial embedding interaction as an example. '$\otimes$' denotes matrix multiplication, and '$\circledast$' denotes 1×1 convolution.

are working for n-th RoI feature.

In order to capture the relation information between different actors, we build an attention module for all RoI features. Because each actor RoI has its own spatial and temporal embedding and the embedding is lighter compared to the feature map, we adopt the attention mechanism [33] between embedding instead of feature maps for efficiency.

Here we remind the general format of multi-head attention [33] we use for exhaustivity. Given a query element and a set of key elements, the multi-head attention module can aggregate the key contents according to the attention weights that measure the compatibility of query-key pairs, adaptively. Formally, let $x = (x_1, \cdots, x_n)$ denote n input elements where $x \in \mathbb{R}^{d_x}$, and the output $z = (z_1, \cdots, z_n)$ where $z_i \in \mathbb{R}^{d_z}$ is computed as a weighted sum of a linearly transformed input:

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V). \tag{1}$$

The weight coefficient α_{ij} is computed by a softmax function:

$$\alpha_{ij} = \frac{\exp z_{ij}}{\sum_{k=1}^{n} \exp z_{ik}}, \quad \text{where} \quad z_i j = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_z}}. \tag{2}$$

We send E^s and E^t into the self-attention module and get the corresponding output ϕ^s, ϕ^t, which have the same shape as the original embeddings E^s, E^t. Then, the final action feature is obtained by:

$$f = \mathcal{G}(\mathcal{F}(f^s, \phi^s), \mathcal{F}(f^t, \phi^t)), \tag{3}$$

where $\mathcal{F}$ is the convolution operation with parameters ϕ and $\mathcal{G}$ is the Spatio-Temporal fusion function. We instantiate $\mathcal{F}$ with 1×1 kernels for efficiency. We experiment with various $\mathcal{G}$ instantiations: summation, concatenation, and cross-attention. We will show detailed effects of these instantiations in the ablation study.

To demonstrate this process, we take the spatial action feature interaction module for example, and illustrate it in Figure 4 in detail. One extra FC layer is employed to obtain the final class prediction logits, of which the dimension is the predefined categories number of one specific dataset.

3.5. Objective Function

Since our proposed model solves the localization and classification in an end-to-end manner, the overall objective function in this work is constituted by two corresponding parts:

$$\mathcal{L} = \underbrace{\lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{L1} \cdot \mathcal{L}_{L1} + \lambda_{giou} \cdot \mathcal{L}_{giou}}_{\text{set prediction loss}} + \underbrace{\lambda_{act} \cdot \mathcal{L}_{act}}_{\text{action}}. \tag{4}$$

The first part is *set prediction loss* [2] which produces an optimal bipartite matching between predictions and ground truth objects. We use $\mathcal{L}_{cls}$ to denote the cross-entropy loss over two classes (containing actor *vs.* not containing actor). The $\mathcal{L}_{L1}$ and $\mathcal{L}_{giou}$ are the box loss driven from [2, 42, 31]. λ_{cls}, λ_{L1} and λ_{giou} are the constant scalars balancing the contributions from these loss terms. As for the second part, $\mathcal{L}_{act}$ is binary cross entropy loss used for action classification, while λ_{act} is the corresponding weight.

4. Experiments on AVA

The AVA [12] is one representative benchmark dataset for testing the spatio-temporal action localization performance. It contains about 211k training clips and 57k validating video clips. The corresponding labels are provided for one frame per second, where every person is annotated with a bounding box and action labels. Following standard protocol [8, 12, 9], we evaluate 60 classes among total 80 classes.

4.1. Implementation Details

Training details. Following previous works [2, 42, 31], we use AdamW [24] with weight decay 0.0001 as the optimizer for all experiments. The mini-batch consists of 16 video clips and all models are trained with 8 GPUs (2 clips per device). Following the $1\times$ training schedule in [31], we train the network for 12 epochs with an initial learning rate of 2.5×10^{-5}. The decay factor is set as 0.1 to decrease the learning rate at epoch 6 and 10, respectively. The backbone is initialized with the pre-trained weights on Kinetics [18] and other newly added layers are initialized with Xavier [11]. We perform random scaling to each of the video frame input, and set its shortest side to range from 256 to 320 pixels and its longest side below 1333 pixels. Following [2, 42, 31], we set the loss weight in person detector head as $\lambda_{cls} = 2$, $\lambda_{L1} = 5$, $\lambda_{giou} = 2$, without extra fine-tuning. Note that further fine-tuning on the target dataset may improve the performance but is out of the scope of this work. For newly added action classification loss, we set the weight as $\lambda_{act} = 4$. The default number of proposal

model	AVA	E2E	$T \times \tau$	pre	val mAP	GFLOPs
AVA baseline [12]	v2.1	✗	64×1	K400	15.6	
Relation Graph [41]			36×1	K400	22.2	
VAT [10]			64×1	K400	25.0	
ACRN [30]			-	K400	17.4	
ATR [16]			-	K400	21.7	
Context-Aware [38]			32×2	K400	28.0	
LFB [37]			32×2	K400	27.6	
X3D-XL [7],			16×5	K400	26.1	
I3D [9]			64×1	K600	21.9	
SlowFast, R50 [8]			8×8	K400	24.7	223.3
SlowFast, R101 [8]			8×8	K600	27.3	302.3
WOO, SFR50		✓	8×8	K400	25.2	141.6
WOO, SFR101			8×8	K600	**28.0**	245.8
SlowOnly, R50 [8]	v2.2	✗	4×16	K400	20.3	136.8
SlowFast, R50 [8]			8×8	K400	24.7	223.3
SlowFast, R101 [8]			8×8	K600	27.4	302.3
WOO, SR50		✓	4×16	K400	21.3	68.0
WOO, SFR50			8×8	K400	25.4	147.5
WOO, SFR101			8×8	K600	**28.3**	251.7

Table 1: **Comparisons with the state-of-the-art methods on AVA V2.1 and V2.2.** 'SR50' denotes 'SlowOnly, R50' backbone variants and 'SFR50' denotes 'SlowFast, R50'. The $T \times \tau$ column shows the frame number and corresponding sample rate. The GFLOPs column contains both actor localization and action classification.

boxes is 100. For efficient exploration, we use the lightweight SlowOnly [8] as the backbone network for ablation studies unless otherwise specified. We also adopt SlowFast R50, R101 as the backbone networks for fair comparisons to the state-of-the-art works.

Inference details. We have a simple inference process in WOO. Given an input video clip, WOO directly predicts 100 bounding boxes associated with their actor detection and action classification scores. The actor detection scores indicate the probability of boxes containing an actor and the action classification scores indicate the probability of every action class to the corresponding box. Furthermore, we only choose the detected boxes with a confidence score larger than 0.7 as the final results.

4.2. Comparisons with State-of-the-Art methods

In Table 1, we compare our method with previous state-of-the-art works on AVA v2.1 (upper part) and V2.2 (lower part) in terms of mAP with IoU threshold 0.5. For fair comparison, we only consider methods using a single model and single cropping for testing. Our proposed approach WOO outperforms the corresponding two-stage, two-backbone counterparts while reducing the model complexity remarkably. Specifically, on AVA v2.1, WOO with SlowFast Res50 video backbone achieves 0.5 mAP gain (25.2 *v.s.* 24.7) while reducing the model complexity by 36.6% (from 223.3 to 141.6 GFLOPs). On AVA v2.2, WOO with SlowFast Res101 backbone achieves 28.3 mAP, 0.9 higher than the counterpart in [8] and 18.7% relative reduction in GFLOPs. We will further detail the complexity versus mAP trade-off in the next section.

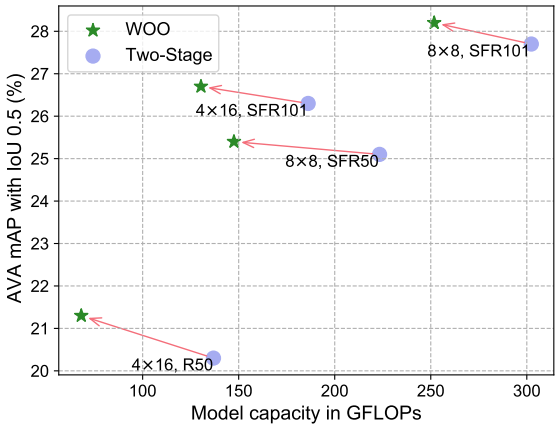

Figure 5: **Model complexity *v.s.* mAP on AVA.** All WOO variants consistently outperform the two-stage baseline counterparts while saving significant GFLOPs.

4.3. Model Complexity versus Accuracy

Figure 5 illustrates the GFLOPs *v.s.* mAP curve for comparisons of WOO variants with different backbones and sample rates over their corresponding two-stage baselines. The horizontal axis measures model capacities of different methods for a single input clip of 320^2 spatial size. The model complexity is computed based on the SlowFast open-source benchmark [6]. Figure 5 shows that all variants of WOO consistently outperform their counterpart baselines for reducing GFLOPs and achieving higher mAP.

4.4. Ablation Study

We perform detailed ablation studies on the AVA dataset to investigate the effects of different components in our model. In addition to the performance metric of mean Average Precision (mAP) with a frame-level Intersection of Union (IoU) threshold of 0.5, we also report the performance of COCO-style AP [22], which averages mAP over different IoU thresholds, from 0.5 to 0.95 (written as AP). The resutls are presented in Table 2. The detailed analysis is as what follows.

Unified person detector. Starting with the SlowFast video backbone, we replace the independent off-the-shelf person detector network with the end-to-end actor detection head used in WOO, and thus, obtain a naïve end-to-end action detection model baseline. This modification reduces the model complexity by over 50% (136.8 to 65.1 GFLOPs), but the action detection performance drops from 20.3 to 16.9 AP_{50}. Interestingly, while actor detection mAP drops a lot, the actor localization performance maintains stable, demonstrating that this performance drop is mainly influ-

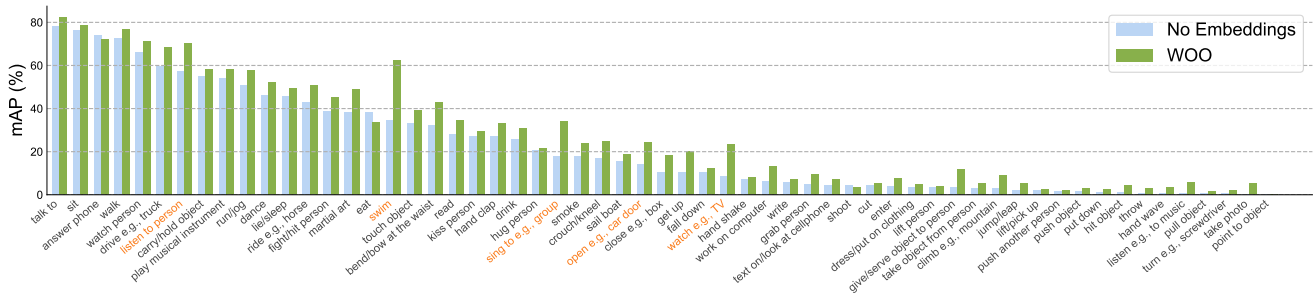

Figure 6: **Per-category AP_{50} on AVA dataset.** Naïvely implemented end-to-end baseline model without embeddings interaction v.s. WOO. The highlighted (orange color) categories are the 5 highest absolute increase. Here we use our best model which have 28.3 mAP. 'No Embeddings' represents WOO w/o interaction embeddings.

E2E	E^s	E^t	AP	AP_{50}	AP_{75}	$AP^{\diamondsuit}_{50}$	GFLOPs
			14.7	20.3	16.6	95.5	136.8
✓			12.3 (−2.4)	16.9 (−3.4)	14.2 (−2.4)	95.5	**65.1**
✓	✓		14.7	20.4 (+0.1)	16.8 (+0.2)	95.6	67.6
✓		✓	13.4 (−1.3)	18.5 (−1.8)	15.4 (−1.2)	95.6	65.8
✓	✓	✓	**15.3** (+0.6)	**21.3** (+1.0)	**17.5** (+0.9)	**95.6**	68.0

Table 2: **Ablation studies on each components in WOO**. '$AP^{\diamondsuit}_{50}$' denotes the image actor detection mAP with threshold of 0.5. A Naïvely implemented unified network (denoted as E2E) reduces model complexity by over 50% but causes a significant performance drop (−3.4). Our spatial and temporal embedding modules (denoted as E^s and E^t) solve the performance drop problem with negligible GFLOPs.

enced by the action classification. Considering that the action classification is more challenging than actor localization [12], we conclude that this naïvely implemented unified framework tends to find an undesired trivial solution (maintaining the easier task performance while hurting more challenging task performance). We strengthen this observation through the next set of ablations. Furthermore, this significant performance drop indicates that building an end-to-end framework for video action detection is not a trivial work.

Spatial and temporal embedding interaction. Based on the naïve E2E model (Row 2 in Table 2), we introduce the spatial and temporal embeddings to make the action classification features more discriminative. The corresponding results are shown in Row 3 and Row 4 of Table 2. Specifically, the spatial embedding improves mAP from 16.9 to 20.4, even achieving a slightly better performance than the two-stage, two-backbone baseline (20.3 mAP). The temporal embedding also brings a mAP gain compared with the naïve E2E model. Finally, using both spatial and temporal embedding, our model (Row 5) achieves +1.0 mAP (from 20.3 to 21.3) gain while reducing nearly by 50% GFLOPs (from 136.8 to 68.0) compared with the baseline (Row 1).

Isolating features using FPN. We design a unified backbone that provides task-specific convolutional features for actor localization and action classification, enabling nearly cost-free feature extraction for actor localization. We implement this unified backbone by adopting the lightweight FPN structure to isolate key frame features starting at the early stage. Compared to the original video backbone, this modification is simple yet effective. Table 3 explores how FPN works. '✗' denotes the use of dilated res5 instead of FPN, which is a default setting in [8].

Model	FPN	Person Detector			AVA			GFLOPs
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	
WOO	✗	74.4	95.3	85.4	14.8	20.8	16.8	90.3
WOO	key	**75.6**	**95.6**	**87.1**	**15.3**	**21.3**	**17.5**	**68.0**
SlowFast	✗				**14.7**	**20.3**	**16.6**	136.9
SlowFast	TP	-	95.5	-	13.8	19.2	15.6	120.4
SlowFast	key				13.7	19.1	15.6	120.4

Table 3: **Effect of FPN.** '✗' denotes the use of dilated res5 instead of FPN following [8]. 'TP' denotes for *temporal pooling* and 'key' for *key frame*. They are two ways to obtain FPN inputs features. See text for detail.

In the upper part of Table 3, we see the actor localization performance is improved remarkably when using FPN, especially measured under the strict metric (+1.2 AP). Moreover, since we replace the dilated res5 used by SlowFast [8] with the FPN structure, the total model complexity is reduced.

For a fair comparison, we also plug the FPN module into the SlowFast baseline and the results are presented in the lower part of Table 3. We select the FPN input features in two ways. 'TP' denotes the use of temporal pooling to reduce the temporal dimension to 1. 'key' denotes the selection of key frame feature. These two implementation methods both damage the overall performance compared to the one using dilated res5. We conclude that for the video backbone that is only responsible for action classification, the FPN structure does not work as well as dilated res5.

Spatial-Temporal Fusion. Table 4 shows various instanti-

No Embeddings: sit[0.92], carry/hold[0.12], touch[0.31] WOO: sit[0.99], carry/hold[0.72], read[0.46]

Figure 7: **Qualitative visualization example.** We visualize the predictions made by WOO with and without embedding interaction. The middle two frames are the key frames predicted by models without (**blue** outline) and with (**yellow** outline) embedding interaction.

ations of fusing temporal and spatial action features, *i.e.*, summation, concatenation and cross-attention (**CA**). The results indicates that CA outperforms summation and concatenation slightly with similar GFLOPs. Thus, we employ the cross-attention fusion operation in all of the WOO variants.

Fusion	AP	AP_{50}	AP_{75}	GFLOPs
sum	14.8	20.5	16.9	68.0
concat	14.7	20.5	17.0	68.1
CA	**15.4**	**21.3**	**17.7**	**68.0**

Table 4: **Spatial-Temporal Fusion.** Fusing spatial and temporal action classification features with various instantiations. cross-attention (CA) achieves a slight better performance.

4.5. Qualitative Results

In Figure 7, we qualitatively present an example to visualize the effect of the embedding interaction. We show the key frame and its surrounding four frames. The middle two frames are the key frames predicted by models without (**blue** outline) and with (**yellow** outline) embedding interaction. As the results show, the model with embedding interaction predict action classes more accurately than the model without embedding. We also compare the performance for every category in Figure 6.

5. Experiments on JHMDB

To verify the effectiveness of our proposed method, we further evaluate our model on the JHMDB dataset. The JHMDB [15] dataset consists of 928 temporally trimmed clips. Every frame in JHMDB contains a single person and has a single action class. JHMDB has 21 action classes and three training/validation splits. Following previous works [38, 19], we report the frame-level mean average precision (frame-mAP) with an intersection-over-union (IoU) threshold of 0.5, over these three splits. The implementation settings are essentially the same as AVA experiments. The results also indicate that our proposed WOO is able to surpass state-of-the-arts on JHMDB dataset.

Main Results. Table 5 shows the results and comparison with previous methods. Our models outperform previous state-of-the-art work. Additionally, we use $T \times \tau = 8 \times 8$, which is less than Context RCNN [38], which uses 16×4. To the best of our knowledge, WOO is the first work that achieves $80+$ mAP with a single model by using only RGB frames on the JHMDB dataset. This outstanding performance demonstrates the generality of our model.

model	$T \times \tau$	flow	pretrain	val mAP
AVA baseline [12]	64×1	✓	K400	73.3
Two-stream RCNN [26]	-	✓	ImgNet	58.5
T-CNN [14]	8×1		UCF101	61.3
TACNet [29]	16	✓	ImgNet	65.5
ACT [17]	6×1		ImgNet	65.7
MOC [20]	7×1		COCO	70.8
P3D-CTN [36]	-		-	71.1
YOWO [19]	16×1		K400	75.7
ACRN [30]	-		K400	77.9
Context-Aware [38]	16×4		K400	79.2
WOO	8×8		K600	**80.5**

Table 5: **Comparison with state-of-the-art on the JHMDB dataset.** Our method achieves 80.5 mAP averaged on three splits, outperforming all published numbers in the literature.

6. Conclusion

We present WOO, an extremely simple end-to-end method for video action detection. It contains a single unified backbone providing task-specific features for actor localization and action classification. Given a video clip, our model directly predicts the bounding boxes and action classes. We validate the proposed method on two challenging video action detection benchmarks and achieve a considerable model complexity reduction compared to start-of-the-art models while achieving a better mAP performance. It is worth noting that our method abandons not only the independent person detector model, but the complex post processing as well. We hope our method will be helpful for the video action detection research community.

Acknowledgements This work was supported by the General Research Fund of Hong Kong No.27208720 and the HKU-TCL Joint Research Centre for Artificial Intelligence grant.

References

[1] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pages 354–370. Springer, 2016. 4

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3, 5

[3] R Christoph and Feichtenhofer Axel Pinz. Spatiotemporal residual networks for video action recognition. *Advances in Neural Information Processing Systems*, pages 3468–3476, 2016. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 4

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 3

[6] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. https://github.com/facebookresearch/slowfast, 2020. 6

[7] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. 2, 3, 4, 6

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019. 2, 3, 4, 5, 6, 7

[9] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018. 5, 6

[10] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019. 6

[11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 5

[12] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018. 2, 5, 6, 7, 8

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[14] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 5822–5831, 2017. 8

[15] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, Dec. 2013. 8

[16] Jianwen Jiang, Yu Cao, Lin Song, Shiwei Zhang4 Yunkai Li, Ziyao Xu, Qian Wu, Chuang Gan, Chi Zhang, and Gang Yu. Human centric spatio-temporal action localization. In *ActivityNet Workshop on CVPR*, 2018. 6

[17] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4405–4413, 2017. 8

[18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 3, 5

[19] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019. 2, 3, 8

[20] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *European Conference on Computer Vision*, pages 68–84. Springer, 2020. 8

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[22] 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. 2, 4, 6

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 4

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[25] Junting Pan, Siyu Chen, Zheng Shou, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. *arXiv preprint arXiv:2006.07976*, 2020. 2

[26] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *European conference on computer vision*, pages 744–759. Springer, 2016. 8

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 3

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 4

[29] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11987–11995, 2019. 8

[30] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018. 2, 6, 8

[31] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020. 3, 4, 5

[32] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 4

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 3, 5

[34] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 4

[35] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[36] Jiangchuan Wei, Hanli Wang, Yun Yi, Qinyu Li, and Deshuang Huang. P3d-ctn: Pseudo-3d convolutional tube network for spatio-temporal action detection in videos. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 300–304. IEEE, 2019. 8

[37] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-Term Feature Banks for Detailed Video Understanding. In *CVPR*, 2019. 2, 3, 4, 6

[38] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware rcnn: A baseline for action detection in videos. In *European Conference on Computer Vision*, pages 440–456. Springer, 2020. 2, 3, 6, 8

[39] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[40] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015. 3

[41] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9975–9984, 2019. 6

[42] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3, 5