Multiscale Vision Transformers meet Bipartite Matching for efficient single-stage Action Localization

Ioanna Ntinou[†,1] Enrique Sanchez[†,2] Georgios Tzimiropoulos[1,2]
[1]Queen Mary University London, UK [2]Samsung AI Center Cambridge, UK
[†]Equal contribution
i.ntinou@qmul.ac.uk kike.sanc@gmail.com g.tzimiropoulos@qmul.ac.uk

Abstract

Action Localization is a challenging problem that combines detection and recognition tasks, which are often addressed separately. State-of-the-art methods rely on off-the-shelf bounding box detections pre-computed at high resolution, and propose transformer models that focus on the classification task alone. Such two-stage solutions are prohibitive for real-time deployment. On the other hand, single-stage methods target both tasks by devoting part of the network (generally the backbone) to sharing the majority of the workload, compromising performance for speed. These methods build on adding a DETR head with learnable queries that after cross- and self-attention can be sent to corresponding MLPs for detecting a person's bounding box and action. However, DETR-like architectures are challenging to train and can incur in big complexity.

In this paper, we observe that ***a straight bipartite matching loss can be applied to the output tokens of a vision transformer****. This results in a backbone + MLP architecture that can do both tasks without the need of an extra encoder-decoder head and learnable queries. We show that a single MViTv2-S architecture trained with bipartite matching to perform both tasks surpasses the same MViTv2-S when trained with RoI align on pre-computed bounding boxes. With a careful design of token pooling and the proposed training pipeline, our Bipartite-Matching Vision Transformer model,* ***BMViT****, achieves +3 mAP on AVA2.2. w.r.t. the two-stage MViTv2-S counterpart. Code is available at https://github.com/IoannaNti/BMViT*

1. Introduction

Action Localization is a challenging task that requires detecting a person's bounding box within a given frame and classifying their corresponding actions. This task shares similarities with object detection, with the particularity that the detected objects are always people and the classes correspond to various actions that can sometimes co-occur. It poses the additional challenge that actions require temporal reasoning, as well as the fact that a person detector can contribute to false positives if a given person is not performing any of the target actions. The current golden benchmark of AVA 2.2. [18] has a fairly low mean Average Precision (mAP) compared to that of e.g. COCO [22].

Interestingly, state-of-the-art approaches achieving high absolute mAP outsource the detection task to a pre-trained Faster-RCNN [27], and focus on large capacity networks and the use of large pre-training data (see Fig. 1a)). These methods serve the purpose of boosting the absolute performance by means of accuracy but are in many cases prohibitive even for standard GPU accelerators.

In this paper, we contribute to the domain of low-regime scalable models that perform both the detection and the recognition tasks. Because of its similarities with the object detection task, many recent methods targeting a single-stage model build on having a strong backbone that provides temporal features to a DETR [1] architecture [44, 47]. A DETR architecture is an encoder-decoder transformer with learnable queries that are assigned to the ground-truth pairs of bounding box/action in a set prediction fashion (Fig. 1b)).

While DETR-based architectures have shown to be efficient choices for end-to-end action localization (i.e. for joint detection and classification tasks), their design involves a video backbone and an encoder-decoder transformer architecture. This poses the question of whether there is room for further improvement in the network design. To answer this question, we draw inspiration from the recent advances in Open-World Object Detection using Vision Transformers (OWL-ViT [24]) and make the following **contribution**: *we propose the use of a bipartite matching loss between the spatio-temporal output embeddings of a single transformer backbone and the ground-truth instances in a video clip* (Fig. 1c). In this setting, the video embeddings are independent tokens that can be matched to the predictions similarly to that of DETR. This implies that a) no learnable tokens are necessary, nor a transformer decoder with self- and cross-

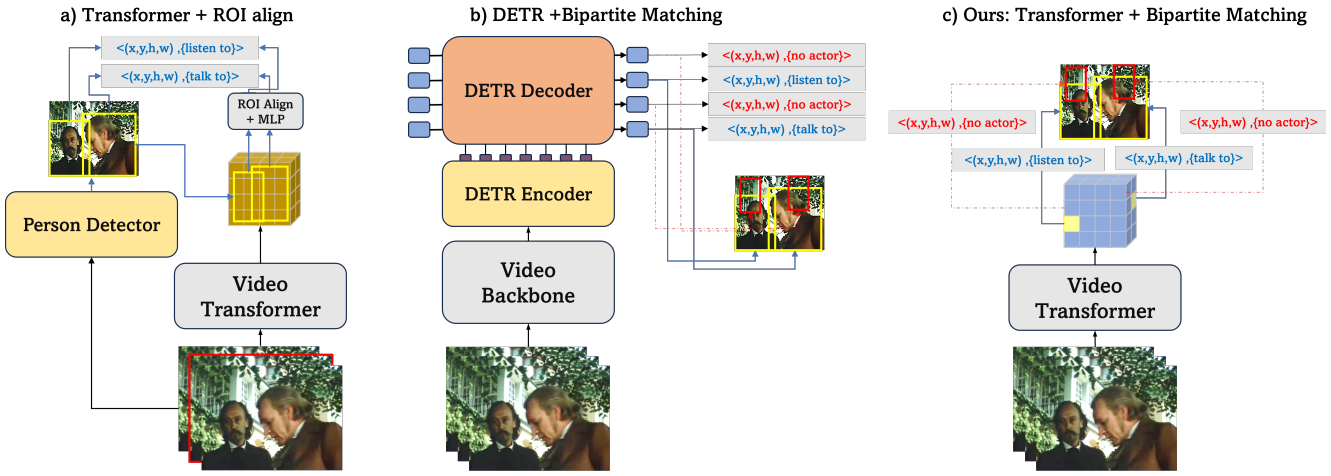

Figure 1. Comparison between existing works and our proposed approach. **a)** Traditional two-stage methods work on developing strong vision transformers, that are applied in the domain of Action Localization by outsourcing the bounding box detections to an external detector. ROI Align is applied to the output of the transformer using the detected bounding boxes, and the pooled features are forwarded to an MLP that returns the class predictions. **b)** Recent approaches in one-stage Action Localization leverage on the DETR capacity to model both the bounding boxes and the action classes. A video backbone produces strong spatio-temporal features that are handled by a DETR transformer encoder. A set of learnable queries are then used by a DETR transformer decoder to produce the final outputs. **c) Our method** builds a vision transformer only that is trained against a bipartite matching loss between the individual predictions given by the output spatio-temporal tokens and the ground-truth bounding boxes and classes. Our method does not need learnable queries, as well as a DETR decoder, and can combine the backbone and the DETR encoder into a single architecture.

attention, and b) the video backbone and the encoder can be merged into a single strong video transformer. Such a simple approach with a careful token selection allows us to train an MViTv2-S [19] with a simple MLP head to directly predict the bounding boxes and the action classes. Without additional elements or data, our single-stage MViTv2-S surpasses the two-stage MViTv2-S of [19] that was trained using RoI align and pre-computed bounding boxes.

Our main results indicate that with similar FLOPs an MViTv2-S trained with bipartite matching performs better than the same MViTv2-S when trained only for action classification from precomputed bounding boxes. In addition, by simply removing the last pooling layer of MViTv2-S we obtain a +3 mAP increase on AVA2.2 [18]. To the best of our knowledge, our method is the first to apply an encoder-only vision transformer for action localization with a bipartite matching loss.

2. Related Work

Two-stage spatio-temporal action localization: Most existing works on action detection [6–8, 25, 36, 39, 41–43, 46] depend on a supplementary person detector for actor localization. Typically that is a Faster RCNN-R101-FPN [27] detector, originally trained for object detection on COCO [20] dataset and subsequently fine-tuned on the AVA [10], is incorporated in the action detection task. By introducing an off-the-self detector, the action detection task is simplified to an action classification problem. For actor-specific prediction, the RoIAlign [12] operation is applied to the generated 3D feature maps. The aforementioned standard pipeline is employed by SlowFast [8], MViT [19], VideoMAE [38] and Hiera [29] where RoI features are directly utilized for action classification. However, such features only confine information within the bounding box, neglecting any contextual information beyond it. To address the limitation, AIA [37] and ACARN [25] employ an additional heavyweight module to capture the interaction between the actor and the context or other actors. Furthermore, to model temporal interactions, MeMViT [42], incorporated a memory mechanism on an MViT [19] backbone. While achieving high accuracy, these methods are inefficient for real-world deployment. Our method is on par with the state-of-the-art MeMViT requiring fewer FLOPs.

Single-stage spatio-temporal action localization: Motivated by the aforementioned limitation of the traditional two-stage pipeline for action detection and classification, several works attempted to tackle both detection and classification in a unified framework. Some works borrow solutions developed for object detection and adapt them to action detection [44, 47] while others simplify training through joint actor proposal and action classification networks [4, 9, 16, 35, 36], or draw their attention to the task of Temporal Action Localization [2, 30, 31, 45]. SE-STAD [35] builds on the Faster-RCNN framework of [27] and incorporates into it the action classification task. Similarly, the Video action transformer network [9] is a transformer-style action detector to aggregate the spatio-temporal context around the target actors.

More recent works [3, 44, 47] leverage on recent advancements of DETR in object detection, and hence, they propose to form the task using learnable queries to model both action and bounding boxes. TubeR [47] proposed a DETR-based architecture where a set of queries, coined Tubelet Queries, simultaneously encode the temporal dynamics of a specific actor's bounding box as well as their corresponding actions. TubeR uses a single DETR head to model the Tubelet Queries, with a classification head that requires an extra decoder for the queries to attend again to the video features. In a similar fashion, DETR-like fashion, STMixer [44] proposes to adaptively sample discriminative features from a multi-scale spatio-temporal feature space and decode them using an adaptive scheme under the guidance of queries. EVAD [3] suggests two video action detection designs: 1) Token dropout focusing on keyframe-centric spatiotemporal preservation and 2) Scene context refinement using ROI align operation and a decoder. Contrary to previous works that use a decoder or a heavy module that introduces interaction features of context or other actors, we demonstrate that a single transformer model trained directly with a bipartite matching can achieve similar accuracy to more complex solutions based on DETR.

3. Method

To motivate our approach, we first depart from a standard application of Vision Transformers for Action Localization with pre-computed bounding boxes, to then provide a brief description of single-stage methods that build on using bipartite matching (i.e. DETR). We then introduce our approach: a Video Transformer with bipartite matching, without learnable queries and decoder.

3.1. Preliminaries

The goal of Action Localization is to detect and classify a set of actions in the central frame I_t of a video clip X composed of T frames. Because not every person in a video clip might be performing an action of interest, we distinguish between a person and an *actor*, i.e. a person doing any of the C target *actions*. An actor at time t is defined by a *bounding box* $\mathbf{b} = [x_c, y_c, h, w]$, with x_c, y_c the normalized coordinates of the box center and h, w the normalized height and width of the box, and an *action class* one-hot vector $\mathbf{a} = \{0, 1\}^C$ representing the activation or not of each class. The action classes do not need to be mutually exclusive (e.g. "talk to" and "point to (an object)" can co-occur).

3.1.1 Vision Transformers for Action Localization

Multi-scale Vision Transformers (MViT, [5, 19]) are self-attention-based architectures that operate on *visual tokens* produced by dividing the input video (or image) into $\tilde{L} = T \times H \times W$ patches of size $3 \times (\tau \upsilon \nu)$, and by projecting each into D-dimensional embeddings through a linear or a convolutional layer. MViT architectures operate hierarchically considering many small input patches of few channels D, progressively increasing the patch size and the channel dimensions through pooling layers. Without loss of generality, we define a (Multi-scale) Vision Transformer as a network $\mathcal{V}$ that produces an output set of $\tilde{L}$ tokens of d dimensions from an input clip $X \in \mathbb{R}^{D \times T \times H \times W}$, as $\tilde{X} = \mathcal{V}(X) \in \mathbb{R}^{\tilde{L} \times d}$, with $\tilde{L} = t \times h \times w$, and $t \leq T$, $w \leq W$, $h \leq H$.

Generally, Vision Transformers are first pre-trained on Action Recognition datasets with video-only mutually exclusive classes, using an additional class token X_0 prepended to X. Then, Vision Transformers are adapted to Action Localization tasks by using an *external* actor detector (i.e. a person detector fine-tuned to return positives on actors only) that provides a bounding box $\hat{\mathbf{b}}$. The output $\tilde{L}$ is then treated as a spatio-temporal feature map: ROI-align is done on the temporally pooled feature map $h \times w$ using $\hat{\mathbf{b}}$, and forwarded to an action classifier to produce the action class probabilities $p(\hat{\mathbf{a}})$.

These models, known as *two-stage*, compromise complexity for accuracy, resulting in solutions with prohibitive complexity. Existing works use a state-of-the-art Faster-RCNN detector to compute the bounding boxes, resulting in models with added complexity of 246 GFLOPs, regardless of the size of the proposed architectures. While MViT has recently been proposed for object detection by adding Mask-RCNN [12] and Feature Pyramid Networks [21], the use of a single architecture to perform both detection and action classification is unexplored. Nonetheless, to our knowledge, ours is the first Vision Transformer that can do bounding box detection and action classification in a single step.

3.1.2 DETR for Action Localization

DETR (DEtection TRansformers [1]) formulate object detection as a bipartite matching problem where a fixed set of L learnable embeddings, known as object queries, are one-to-one matched to a list of predictions of the form $\langle \mathbf{b}, p(\hat{\mathbf{a}}') \rangle$, with $\hat{\mathbf{a}}' = \{\hat{\mathbf{a}}, \varnothing\}$ the list of C target classes (objects in this case) appended with the empty class $\varnothing$ representing a no-object class. During training, the Hungarian Algorithm [17, 34] is used to map the $N < L$ objects in an image to the "closest" predictions so that the assignments minimize a combined bounding box and class cost. The $L - N$ remaining predictions are assigned to the empty class. The learning is done by backpropagating the bounding box and class error for those outputs matched to a ground-truth object, and only the class error for those assigned to the empty class (i.e. to enforce bounding boxes that do not correspond to an object to be tagged as $\varnothing$).

DETR possesses appealing properties for Action Localization, as the learnable object queries can convey object lo-

calization and action classification. The handful of proposed approaches that have ventured to apply DETR to Action Localization are faithful to the DETR configuration [44, 47]: a 3D backbone produces video features that are fed into a transformer encoder to produce a hierarchy of features, and a transformer decoder transforms the learnable queries through self- and cross-attention layers between the queries and the encoder features, producing a fixed set of L outputs. However, rather than appending the target classes with the empty class, a dedicated *actor detector* head is used to determine if a bounding box corresponds to an actor or not. The outputs are now triplets of the form $\langle \mathbf{b}, p(\alpha), p(\hat{\mathbf{a}}) \rangle$, with $p(\alpha)$ being a two-logit vector representing the actor and the $\varnothing$ classes. The use of a DETR encoder-decoder architecture offers a good tradeoff between accuracy and complexity with the decoder alleviating part of the backbone's complexity. However, these approaches still suffer from a complex architecture and a limited number of queries that can be used to train the models.

3.2. Our solution: Bipartite-Matching ViT

Herein, we observe that neither the backbone nor the decoder are necessary to achieve a good accuracy-speed tradeoff, and we follow the recent advances in Open-Vocabulary Object Detection(OWL-ViT [24]) to motivate our approach. OWL-ViT introduces a CLIP vision transformer encoder with the last pooling layer removed. Instead of pooling the output tokens to form a visual embedding to be mapped into the text embeddings as in the standard CLIP image-text matching, each of the output embeddings is forwarded to a small head consisting of a bounding box MLP and a class linear projection. This way, the $\tilde{L} = h \times w$ output tokens from the vision encoder are treated as independent output pairs $\langle \hat{\mathbf{b}}, p(\hat{\mathbf{a}}) \rangle$. These pairs are matched to the ground truth using DETR's bipartite matching loss. In OWL-ViT, the image patches play the role of the object queries.

To adapt OWL-ViT to the domain of Action Localization, we first note that Multi-scale Vision Transformers are a natural pool of spatio-temporal output embeddings that can be one-to-one matched to triplets $\langle \hat{\mathbf{b}}, p(\alpha), p(\hat{\mathbf{a}}) \rangle$. As we demonstrate in §5.1, a simple MViTv2-S architecture trained with bipartite matching achieves higher accuracy than the same MViTv2-S trained for Action Localization using external bounding boxes. Notably, our approach, which we coin **Bipartite-Matching Vision Transformer**, or **BMViT**, does not add additional complexity to the backbone, given that the heads are simple MLPs.

The output of the video transformer, as introduced in §3.1.1, is $\tilde{X} = \mathcal{V}(X) \in \mathbb{R}^{\tilde{L} \times \tilde{D}}$, with $\tilde{L} = t \times h \times w$ corresponding to the output sequence length. For instance, MViTv2-S produces, for an input video of $16 \times 256 \times 256$, an output of $8 \times 8 \times 8$ tokens, i.e. $\tilde{L} = 512$, which outnumbers DETR-based architectures. The number of output tokens

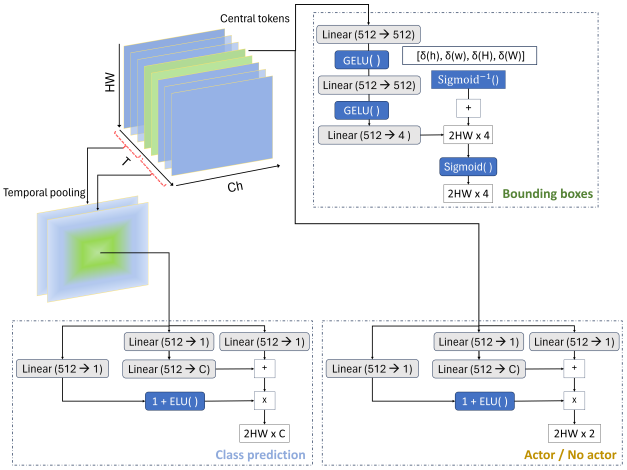

Figure 2. The output spatio-temporal tokens are fed to 3 parallel heads. We use the central tokens to predict the bounding box and the actor likelihood while averaging the output tokens over the temporal axis to generate the action tokens. Each head comprises a small MLP that generates the output triplets. We depict the flow diagram for each head, following the standard OWL-ViT head [24].

does not affect the complexity of the network as these are independently processed by three MLPs. Following OWL-ViT, we add a bias to the predicted bounding boxes to make each be centered by default on the image patch that corresponds to the 2D grid in which the output tokens would be re-arranged. As reported in [24], "there is no strict correspondence between image patches and tokens representations"; however, "biasing box predictions speeds up training and improves final performance". In our 3D space-time setting, we add the same 2D bias to all tokens on the same 2D grid along the temporal axis.

Because each head will process the output tokens independently to produce $\hat{\mathbf{b}}$, $p(\alpha)$ and $p(\hat{\mathbf{a}})$, we can select which of these are of better use to the detection and recognition subtasks. *The only technical limitation is that the outputs of each head need to be in one-to-one correspondence with those from the other heads, to form the triplets that will be matched to the ground-truth instances*. This is an important consideration because the tasks of actor detection and action classification are opposed by definition: while detecting actors requires only information from the central frame, the task of action recognition benefits from using temporal support. As we are not limited to use the same output tokens for each task, we can consider e.g. the $w \times h$ tokens corresponding to $t = \lfloor T/2 \rfloor$ to generate $\tilde{L} = w \times h$ bounding boxes $\hat{\mathbf{b}}$ and actor/no actor probabilities $p(\alpha)$, and apply temporal pooling to produce an equivalent set of $\tilde{L}$ tokens that will be used to compute the action probabilities $p(\hat{\mathbf{a}})$.

Bearing this in mind, we can handle the visual tokens in a way that benefits both tasks. First, to avoid poor detection when using MViTv2-S, we remove the last pooling layer to increase the output resolution, i.e. to produce an output of

$\tilde{L} = 8 \times 16 \times 16 = 2048$ tokens. This way, we can use the $\tilde{L} = h \times w = 256$ central tokens for the detection task, and apply temporal pooling to form the equivalent $\tilde{L} = 256$ spatio-temporal tokens to be used for the classification task. We also study an alternative strategy that considers, for the bounding box and actor/no actor predictions, the tokens corresponding to both $t = \lfloor T/2 \rfloor$ *and* $t = \lceil T/2 \rceil$. Concatenating both results in $\tilde{L} = 512$ tokens to perform the actor detection task. To produce the corresponding $\tilde{L} = 512$ "action" tokens, we apply temporal pooling on the past and future tokens w.r.t. the central frame, independently. That is to say, we compute $\tilde{X}'_{t<T/2} \in \mathbb{R}^{h \times w}$ as $\tilde{X}'_{t<T/2\,[i,j]} = (2/T) \sum_{t'<T/2} \tilde{X}_{t',i,j}$ for $i, j \in [1...h, 1...w]$, and similarly $\tilde{X}'_{t>T/2} \in \mathbb{R}^{h \times w}$. We then concatenate both to form the final set of $\tilde{L} = 512$ tokens. Note that both the "actor" and "action" tokens have a clear one-to-one correspondence, which is the necessary condition to produce $\tilde{L}$ triplets $\langle \tilde{\mathbf{b}}, p(\alpha), p(\tilde{\mathbf{a}}) \rangle$. As we observe in §5.2, this approach results in better performance than the one above that considers only $\tilde{L} = 256$ tokens. Notably, this strategy also outperforms that of considering the full set of $\tilde{L} = 2048$ tokens, pointing to the need of carefully choosing which tokens are more suitable for each subtask.

Training During training, the output tokens are forwarded to the corresponding heads to predict the $\tilde{L}$ triplets $\hat{y}_l = \langle \hat{\mathbf{b}}_l, p(\alpha)_l, p(\hat{\mathbf{a}})_l \rangle$ with $l \in [0, \tilde{L} - 1]$. We use the Hungarian Algorithm to match these predictions to the ground-truth instances. A ground-truth instance with N bounding boxes is defined as $y_j = \langle \mathbf{b}_j, \alpha_j = 1, \mathbf{a}_j \rangle$ for $j < N$, and as $y_j = \langle \mathbf{b}_j = \varnothing, \alpha_j = \varnothing, \mathbf{a}_j = \varnothing \rangle$. The matching cost between a prediction $\hat{y}_i$ and a ground-truth triplet y_j is defined as

$$\begin{aligned} \mathcal{C}(\hat{y}_i, y_j) = \mathbb{1}_{[\alpha=1]} \mathcal{L}_{box}(\mathbf{b}_j, \hat{\mathbf{b}}_i) - \\ \mathbb{1}_{[\alpha=1]} \mathcal{L}_{actor}(\alpha_j, p(\alpha_i)) - \mathbb{1}_{[\alpha=1]} \mathcal{L}_{class}(\mathbf{a}_j, p(\mathbf{a}_i)). \end{aligned} \tag{1}$$

with $\mathcal{L}_{box}$ the bounding box loss, $\mathcal{L}_{actor}$ the actor/no actor loss, and $\mathcal{L}_{class}$ the classification loss, respectively. Following DETR [1] the bounding box loss is defined as $\mathcal{L}_{box}(\mathbf{b}, \hat{\mathbf{b}}) = \mathcal{L}_{iou}(\mathbf{b}, \hat{\mathbf{b}}) + \|\mathbf{b} - \hat{\mathbf{b}}\|_1$, with $\mathcal{L}_{iou}$ the generalized IoU loss [28]. The actor/no actor loss is defined as $\mathcal{L}_{actor}(\alpha_j, p(\alpha_i)) = -\alpha_j \log p(\alpha_i) + (1 - \alpha_j) \log(1 - p(\alpha_i))$, and the classification loss as:

$$\begin{aligned} \mathcal{L}_{class}(\mathbf{a}_j, p(\mathbf{a}_i)) = -\sum_c \mathbf{a}_{j,c} \log p(\mathbf{a}_{i,c}) \\ + (1 - \mathbf{a}_{j,c}) \log(1 - p(\mathbf{a}_{i,c})), \end{aligned} \tag{2}$$

where $\mathbf{a}_i = \{\mathbf{a}_{i,c}\}_{c=1...C}$ is a C-d vector of 1s and 0s representing the multi-label nature of the problem, and $p(\mathbf{a}_{,c})$ represents the model confidence for class c. For $\alpha = 0$, we simply set $\mathbf{a}_{i,c} = 0$, $\forall c$. Note that Eq. (1) only considers the cost of the assignments w.r.t. the annotated bounding boxes (i.e. $\alpha = 1$).

Once an optimal assignment $i = \sigma(j)\ \forall i \in [0, ..., \tilde{L} - 1]$ is found, we compute the *Hungarian Loss* as:

$$\begin{aligned} \tilde{\mathcal{L}} = \sum_{j=1}^{L} [\mathbb{1}_{[\alpha=1]} \left(\lambda_{iou} \mathcal{L}_{iou}(\mathbf{b}, \hat{\mathbf{b}}) + \lambda_{L1} \|\mathbf{b} - \hat{\mathbf{b}}\|_1 \right) - \\ \lambda_\alpha \mathcal{L}_{actor}(\alpha_j, p(\alpha_\sigma(j))) - \lambda_a \mathcal{L}_{class}(\mathbf{a}_j, p(\mathbf{a}_\sigma(j)))] \end{aligned} \tag{3}$$

where $\lambda = \{\lambda_{iou}, \lambda_{L1}, \lambda_\alpha, \lambda_a\} \in \mathbb{R}^4$ are hyperparameters.

Inference During inference, we compute the L output triplets, and we simply keep the detections of those for which $p(\alpha) > \theta$ with θ a hyperparameter that controls the trade-off between precision and recall.

Remarks We want to highlight that our method, which resides in reformulating the training objective of vision transformers for action localization and in the observation that such approach comes with different alternatives for token selection, is amenable to different backbones, token selection design, and even output resolution. The fact that tokens are fixed and assigned to ground-truth instances also implies that we can directly resize the input frames to a fixed squared resolution, without any concern regarding losing the aspect ratio, something not possible in ROI-based approaches. This prevents our method from the need of using different views, being computationally more efficient.

4. Experimental Setup

Datasets. We use **AVA 2.2** [10, 18] to conduct our main experiments and ablations, and **UCF101-24** [33] and **JHMDB51-21** [13] to demonstrate the generalising capabilities of our approach. **AVA 2.2** [10, 18] is a long-tail dataset with 299 videos of 15-minute duration, annotated with bounding boxes and 80 action classes at 1 FPS rate. The training and validation partitions contain 235 and 64 videos, amounting to 211k and 57k frames, respectively. We follow the standard evaluation protocol and report our results on the 60-class subset of annotated actions [6, 8, 25, 47]. **UCF101-24** [33] contains 3207 untrimmed videos, and contains box labels and annotations on a per-frame basis, with 24 classes. We follow the standard protocol defined in [32] and report our results on split-1. **JHMDB51-21** [13] has 928 trimmed videos that are labeled with 21 action categories. We follow prior work and report the average results on the three splits. For all datasets, we report the standard mean average precision (mAP) computed considering as false positives all predictions corresponding to bounding boxes with IoU < 0.5 w.r.t. a ground-truth box.

Implementation details: We initialize our model weights from the publicly available checkpoint of [19] pretrained on Kinetics-400 [15]. The network architecture and the output sizes are summarized in the Supplementary Material. The

Setting ($t \times hw$)	GFLOPs	mAP
MViTv2-S [19]	64 + 246	26.8
Inp: 16×224^2, Out: 8×7^2	**65**	26.8
Inp: 16×224^2, Out: 8×14^2	87.9	27.4
Inp: 16×256^2, Out: 8×8^2	90.7	27.5
Inp: 16×256^2, Out: 8×16^2	121.2	**30.0**

(a) vs MViT + ROI align

Method	mAP	Prec.	Recall
Singleton	28.5	84.2	90.4
Tubelet	28.7	83.2	90.5
C + T	29.1	81.1	92.7
Max Pooling	28.3	84.1	91.3
2(C+T)	**30.0**	80.0	92.0

(b) Token Selection

Method	mAP
Variable aspect ratio	28.5
Fixed aspect ratio	**30.0**

(c) Impact of aspect ratio

Table 1. Ablation studies on AVA 2.2 All experiments are done using an MViTv2-S [19] pre-trained on K400. **a)** We study the relation between complexity and input/output resolutions, by removing or not the last pooling stride. Note that MViTv2-S [19] requires external bounding boxes, reportedly adding 246 GFLOPs to the overall inference. **b)** We study the impact of different methods for token selection to perform the bipartite matching. **c)** We study whether keeping the input aspect ratio affects the training, observing that a variable number of tokens results in difficult convergence.

input to the model is $T = 16$ frames sampled at a stride of $\tau = 4$. All experiments are done using PyTorch [26]. We train our models using AdamW [23] with weight decay 0.0001, and set the trade-off hyperparameters in Eq. (3) to $\lambda_\alpha = 2.0$, $\lambda_a = 6.0$ $\lambda_{L1} = 5.0$, and $\lambda_{iou} = 2.0$. We use a batch size of 16 clips, and we train our models using 8 GeForce 3090 GPU cards. We train our model for 25 epochs with initial learning rate of 0.0001 and cosine decay. During training, we resize the videos to 256 pixels without cropping. We add jitter to the ground-truth bounding boxes and apply color augmentation. We report our results using a single view with images directly resized to 256×256 pixels.

5. Ablation studies

5.1. vs MViT + ROI align

We first show that our method offers significant improvement w.r.t. the standard two-stage approach of MViT. Because removing the last pooling stride of our model increases the number of FLOPs, we also train a model that preserves the original MViT structure. In such a scenario, the output is a volume of $t = 8$, $h = w = 8$ for an input resolution of 256×256. In the MViT + ROI align setting [19] a temporal pooling layer is added to remove the time dimension, and ROI align is used to extract the actor features from precomputed bounding boxes. The actor-specific features are forwarded to a classifier to predict the class probabilities. In our setting, we generate a set of 64 triplets consisting of bounding box coordinates with their corresponding actor likelihoods and class probabilities. We keep the predictions corresponding to the tokens for which the actor likelihood is over a threshold empirically set to be $\theta = 0.2$. We also consider reducing the FLOPs by working at a lower resolution and study how our method works at a resolution of 224 pixels. The output number of tokens when working at 224 resolution reduces to 14×14. We finally explore the scenario where the resolution is dropped to 224, and the pooling layers are left as in the original MViTv2. Such a model produces only 7×7 tokens at the central frame. The results are summarized in Tab. 1a.

5.2. Token Selection

As mentioned in §3.2, the token selection design results in a different number of tokens to be assigned to the ground-truth. In this Section, we study how such selection affects performance. For an MViTv2-S without the last pooling layer, this results in $\tilde{L} = 2048$ tokens. In §3.2 we introduced two alternatives to the token selection designs, namely that of considering the central frame only for detection and temporal pooling for classification, which we refer to as **C+T**, and that depicted in Fig. 2, which we denote as **2(C+T)**. The former produces $\tilde{L} = 256$ tokens, whereas the latter produces $\tilde{L} = 512$ tokens. In addition, we study three more alternatives, namely **singletons**, where we directly consider the $\tilde{L} = 2048$ tokens without further reduction; **tubelets**, whereby we directly apply temporal pooling to the output embeddings of the MViTv2-S backbone, producing the same $\tilde{L} = h \times w = 256$ for both the detection and recognition tasks; and **max-pooling of class predictions** which considers the central tokens for the actor detection task, and a max-pooling operation on the temporal domain over the outputs of the $\tilde{L} = 2048$ possible action tokens. The results in Tab. 1b indicate that a proper token selection is important to achieve a good tradeoff between detection and classification. We observe that two factors affect the most to improve the mAP: a high recall in the bounding box detection and a good selection of representatives for action tokens.

5.3. Fixed vs Variable Aspect Ratio

Often, detection frameworks apply a scale augmentation by resizing the images to different scales, and by keeping the aspect ratio in the case of transformers. Such augmentation incurs a variable number of tokens $\tilde{L}$ to be assigned for each clip, which might affect the learning. We compare both approaches by training our model using a scale range between 240 and 340 pixels and keeping the aspect ratio.

As shown in Tab. 1c, the performance drops significantly with respect to using a fixed number of tokens. We attribute this effect of matching a variable number of tokens to the ground-truth during training, which might require further hyperparameter optimization to improve convergence. We leave for future work improving the training in the case of variable-size images.

5.4. Qualitative analysis

In Fig. 3 we show a visual demonstration of how the tokens carry the actor information properly. The left images show the confidence maps (i.e. $p(\alpha)$) for each of the $h \times w$ tokens at $t = \lfloor T/2 \rfloor$ (for the sake of clarity we illustrate only the confidence maps for the frame where the actor confidences were resulting in positive detections). We can see that the tokens around the actors in the frame are more confident than those that are farther away. In the right images, we overlap all the bounding boxes computed for the same $h \times w$ tokens, representing in yellow those corresponding to the activated tokens. We can see that while there are other bounding boxes around the two actors, only those that are maximally overlapping the ground-truth activate the actor likelihood with high confidence. We include more qualitative results in the Supplementary Material.

6. Comparison with State-of-the-art

6.1. Performance on AVA 2.2

We compare our method against current state-of-the-art approaches in AVA 2.2, as summarized in Tab. 2. We use symbols ✗ to denote methods relying on pre-computed bounding boxes, indicating a two-stage process, and symbols ✓ for those employing a single-stage approach. Additionally, we provide insights into the computational costs associated with each method. Note that two-stage methods listed in Tab. 2 utilize bounding boxes generated by [8], i.e. they operate a FasterRCNN-R101-FPN network that requires 246 GFLOPs on an input resolution of 512 pixels [1].

Comparison with models pre-trained on K400: We are mostly interested in attaining increased accuracy at a low computational cost. When comparing our method to two-stage approaches, we note that our method using MViTv2-16 $\times$ 4 as the backbone surpasses all MViT methods, and achieves higher accuracy than MeMViT, with fewer FLOPs.

Comparison with recent single stage approaches: We compare our method against TubeR [47], STMixer [44] and EVAD [3] which are the latest works on single-stage action detector. For a fair comparison, and given that we have trained on models initialized on K400, we do not report and compare large models pre-trained on K700. In K400 we achieve 30.0 mAP comparable to TubeR and STMixer. However, our method offers a simpler solution without explicit context modeling and without the use of a decoder.

Figure 3. Qualitative analysis. The images on the **left** show the confidence maps produced by the output 16×16 spatial tokens (rescaled to the image size) w.r.t. the actor likelihood for the corresponding bounding box. For the sake of clarity, we only plot the 256 tokens corresponding to one of the frames. The highlighted tokens are those selected as positive detections. The images on the **right** show all the bounding boxes computed by the corresponding tokens on the left. We overlay all the bounding boxes returned by each of the 16×16 output tokens. In yellow we represent the bounding boxes corresponding to the confident tokens represented on the left. All other bounding boxes (in red) are assigned to the no-class label $\varnothing$, and are thus considered as negative predictions

Generalization To further demonstrate the capabilities of our method, we trained the same ViT-B backbone used for EVAD [3], pre-trained on K400 and VideoMAE [38], without token dropout. Following EVAD, we trained our model using a resolution of 288 pixels. For this model, we used only the central tokens for both actor detection and action classification, resulting in $\tilde{L} = 18 \times 18 = 324$ output tokens. We added a cross attention layer to the action classification head to account for the temporal information, whereby the $\tilde{L}$ output tokens first attend all the $8 \times 18 \times 18 = 2592$ spatio-temporal tokens before being forwarded to the action classification MLP. Note that this cross-attention layer

Method	Pretraining	mAP	GFLOPs	Res.	Backbone	End-to-end
ACAR-Net [25]	K400	28.8	$205 + 246$	256	SF-R50-NL	✗
MViTv1-B [6]	K400	27.3	$455 + 246$	224	MViTv1-B	✗
MViTv2-S [19]	K400	26.8	$65 + 246$	224	MViTv2-S	✗
MViTv2-B [19]	K400	28.1	$225 + 246$	224	MViTv2-B	✗
MeMViT [42]	K400	28.5	$59 + 246$	224	MViTv1	✗
WOO [4]	K400	25.4	148	256	SF-R50	✓
STMixer [44]	K400	27.8	N/A	256	SF-R50	✓
Ours (BMViT)	K400	**30.0**	**121**	256	MViTv2-S 16×4	✓
VideoMAE [38]	K400, MAE	31.8	$180 + 246$	224	ViT-B	✗
EVAD [3]	K400, MAE	**32.1**	425	288	ViT-B	✓
Ours (BMViT)	K400, MAE	31.4	**350**	288	ViT-B 16×4	✓

Table 2. Comparison w.r.t. state-of-the-art (reported with mean Average Precision; mAP ↑) on AVA v2.2 [10]. "Res." denotes frame resolution.

(with 324 queries and 2592 keys and values), amounts to only 4.7 GFLOPs, maintaining a computationally efficient action head. The results in Tab. 2 (bottom) show that our method offers a similar complexity/performance tradeoff w.r.t. EVAD.

Scaling up to large backbones: While the scope of our proposed approach is to develop efficient methods for training a video transformer for action localization, we explore how our proposed approach scales to larger models. To this end, we train a Hiera-L [29] in an end-to-end fashion using the publicly available checkpoint pre-trained on Kinetics400, which was first pre-trained in a self-supervised manner using Masked AutoEncoders [11]. We directly reuse our training recipe to train the large model, without any further parameter optimization. We observe that our approach competes with the two-stage approach of Hiera-L [29], although results in sub-par performance. While Hiera-L obtains 39.8 mAP in a two-stage approach incurring in $413 + 246$ GFLOPs, our single-stage equivalent obtains 38.5 mAP with 650 GFLOPs (full results are included in the Supplementary Material). We attribute this sub-par performance to the difficulty of training large models in an end-to-end fashion. We leave for future research the optimization of big models in our setting.

6.2. Additional Datasets

To demonstrate the effectiveness of our approach we evaluate our method on **UCF101-24** [33] and **JHMDB51-21** [13] datasets. Results of our approach are shown in Tab. 3. We note that compared to EVAD [3] we have similar performance with a backbone of much smaller capacity, i.e. our backbone has 121 GFLOPs while EVAD [3] has 243 GFLOPS. Our approach surpasses EVAD on UCF24, and lies behind in JHMDB. However, for JHMDB, we observed that our model attains precision and recall of of 95% and 97% correspondingly, indicating that further attention needs to be put towards improving accuracy.

Method	End-to-end	Backbone	JHMDB	UCF24
ACAR-Net [25]	✗	SF-R50	-	84.3
ACT [14]	✓	VGG	65.7	69.5
MOC [18]*	✓	DLA32	70.8	78.0
ACRN [36]	✓	S3D-G	77.9	-
YOWO [16]	✓	3D-X101	80.4	74.4
WOO [4]	✓	SF-R101-NL	80.5	-
TubeR [47]*	✓	I3D	80.7	81.3
TubeR [47]	✓	CSN-152	-	83.2
STMixer [44]	✓	SF-R101-NL	**86.7**	83.7
Ours (BMViT)	✓	MViTv2-S	80.7	**85.6**
EVAD [3] @ 288	✓	ViT-B	**90.2**	85.1
Ours (BMViT) @ 288	✓	ViT-B	85.4	**87.3**
Ours (BMViT) @ 288	✓	ViT-B†	88.4	**90.7**

Table 3. Comparison with the state-of-the-art on UCF101-24 and JHMDB. ✓ denotes an end-to-end approach using a unified backbone, and ✗ denotes the use of two separated backbones, one of which is Faster R-CNN-R101-FPN (246 GFLOPs [27]) to pre-compute person proposals. $T \times \tau$ refers to the frame number and corresponding sample rate. Methods marked with * leverage optical flow input. †: ViT-B backbone pre-trained on K710 and fine-tuned on K400 from VideoMAE V2 [40].

7. Conclusion

In this paper, we presented a simple method for the direct training of Vision Transformers for end-to-end action localization. We showed that the output tokens of a vision transformer can be independently forwarded to corresponding MLP heads to have a fixed sequence of predictions similar to DETR. Using a bipartite matching loss it is possible to train the backbone directly to perform both tasks without compromising performance. Our results show that simple models achieve similar accuracy to equivalent two-stage approaches.

Acknowledgments Ioanna Ntinou is funded by Queen Mary Principal's PhD Studentships. This research utilized Queen Mary's Apocrita HPC facility, supported by QMUL Research-IT. http://doi.org/10.5281/zenodo.438045.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 3, 5, 7

[2] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster R-CNN architecture for temporal action localization. In *CVPR*, 2018. 2

[3] Lei Chen, Zhan Tong, Yibing Song, Gangshan Wu, and Limin Wang. Efficient video action detection with token dropout and context refinement. In *ICCV*, 2023. 3, 7, 8

[4] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *ICCV*, 2021. 2, 8

[5] 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. In *ICLR*, 2021. 3

[6] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021. 2, 5, 8

[7] Gueter Josmy Faure, Min-Hung Chen, and Shang-Hong Lai. Holistic interaction transformer network for action detection. In *WACV*, 2023. 2

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2, 5, 7

[9] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019. 2

[10] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 2, 5, 8

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2021. 8

[12] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 3

[13] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013. 5, 8

[14] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017. 8

[15] 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*, 2017. 5

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

[17] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955. 3

[18] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv*, 2020. 1, 2, 5, 8

[19] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MViTv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022. 2, 3, 5, 6, 8

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *arXiv*, 2014. 2

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[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 *ECCV*, 2014. 1

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv*, 2017. 6

[24] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. In *ECCV*, 2022. 1, 4

[25] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *CVPR*, 2021. 2, 5, 8

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 6

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2, 8

[28] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 5

[29] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *arXiv*, 2023. 2, 8

[30] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017. 2

[31] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016. 2

[32] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017. 5

[33] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. 5, 8

[34] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *CVPR*, 2016. 3

[35] Lin Sui, Chen-Lin Zhang, Lixin Gu, and Feng Han. A simple and efficient pipeline to build an end-to-end spatial-temporal action detector. In *WACV*, 2022. 2

[36] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018. 2, 8

[37] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *ECCV*, 2020. 2

[38] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 2, 7, 8

[39] Oytun Ulutan, Swati Rallapalli, Carlos Torres, Mudhakar Srivatsa, and B Manjunath. Actor conditioned attention maps for video action detection. In *WACV*, 2020. 2

[40] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *CVPR*, 2023. 8

[41] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 2

[42] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *CVPR*, 2022. 2, 8

[43] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware RCNN: A baseline for action detection in videos. In *ECCV*, 2020. 2

[44] Tao Wu, Mengqi Cao, Ziteng Gao, Gangshan Wu, and Limin Wang. Stmixer: A one-stage sparse action detector. In *CVPR*, 2023. 1, 2, 3, 4, 7, 8

[45] Ke Yang, Peng Qiao, Dongsheng Li, Shaohe Lv, and Yong Dou. Exploring temporal preservation networks for precise temporal action localization. In *AAAI*, 2017. 2

[46] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *CVPR*, 2019. 2

[47] Jiaojiao Zhao, Xinyu Li, Chunhui Liu, Shuai Bing, Hao Chen, Cees GM Snoek, and Joseph Tighe. Tuber: Tube-transformer for action detection. *arXiv*, 2021. 1, 2, 3, 4, 5, 7, 8