Action Tube Generation by Person Query Matching for Spatio-Temporal Action Detection

Kazuki Omi, Jion Oshima and Toru Tamaki[a]
Nagoya Institute of Technology, Japan
{*k.omi.646, j.oshima.204*}*@nitech.jp, tamaki.toru@nitech.ac.jp*

Keywords: Spatio-Temporal Action Detection (STAD), Action Tubes, Query Matching, DETR, Query-Based Detection, IoU-Based Linking.

Abstract: This paper proposes a method for spatio-temporal action detection (STAD) that directly generates action tubes from the original video without relying on post-processing steps such as IoU-based linking and clip splitting. Our approach applies query-based detection (DETR) to each frame and matches DETR queries to link the same person across frames. We introduce the Query Matching Module (QMM), which uses metric learning to bring queries for the same person closer together across frames compared to queries for different people. Action classes are predicted using the sequence of queries obtained from QMM matching, allowing for variable-length inputs from videos longer than a single clip. Experimental results on JHMDB, UCF101-24 and AVA datasets demonstrate that our method performs well for large position changes of people while offering superior computational efficiency and lower resource requirements.

1 INTRODUCTION

In recent years, the importance of not only image recognition, but also video recognition, especially the recognition of human actions, has been increasing in various practical applications. Among the tasks that recognize human actions in videos, *spatial-temporal action detection* (or *STAD*) (Chen et al., 2023; Sun et al., 2018; Wu et al., 2019; Chen et al., 2021; Li et al., 2020; Zhao et al., 2022; Gritsenko et al., 2023), which detects the class, location, and interval of actions occurring in the video, is important in practical applications (Ahmed et al., 2020).

The goal of STAD is to create *action tubes* (or simply *tubes*) (Gkioxari and Malik, 2015). A tube is a sequence of bounding boxes of the same action class for the same person across the frame of the video. Here, a *bounding box* (or *bbox*) indicates the rectangle enclosing the actor in the frame, and a *tubelet* links bounding boxes of the same action class for the same person within a video clip, a short segment of the original video consisting of several (8 or 16) frames, due to the manageability for video recognition models. A *tube*, on the other hand, links these boxes throughout the video. Thus, a tube provides spatio-temporal information for understanding the action's temporal progression over the frames and its spatial location within the frames.

Although the length of the clips and the architecture of the models varies, most prior works (Köpüklü et al., 2021; Chen et al., 2021; Kalogeiton et al., 2017; Zhao et al., 2022; Gritsenko et al., 2023) create tubes through the following post-processing (see Figure 1a). First, the original video is divided into multiple clips. Next, the model outputs a bounding box (bbox) or tubelet from the clip input. The model outputs are then linked using a linking algorithm (Singh et al., 2017; Li et al., 2020) based on Intersection-over-Union (IoU) to create tubes.

However, there are two disadvantages to the post-processing of creating a tube by linking bbox or tubelets. The first is IoU-based linking. It cannot handle large or fast motion of actors, and significant movements due to rapid camera motion or low fps (Singh et al., 2023). For example, in actions with large fast movements such as "diving" and "surfing," the IoU with adjacent frames may be small. In cases where the camera moves significantly, such as with in-car cameras rather than fixed cameras like surveillance cameras, even actions with small movements may have large displacements between frames, resulting in a small IoU. Thus, the types of actions and camera environments that can be linked by IoU are limited.

[a] https://orcid.org/0000-0001-9712-7777

Omi, K., Oshima, J. and Tamaki, T.
Action Tube Generation by Person Query Matching for Spatio-Temporal Action Detection.
DOI: 10.5220/0013089500003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 261-268
ISBN: 978-989-758-728-3; ISSN: 2184-4321

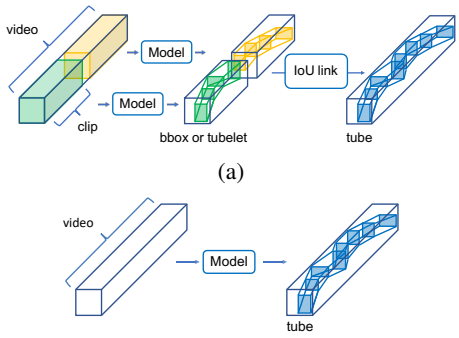

Figure 1: How to create action tubes. (a) A typical process of prior works. First, the original video is divided into multiple clips. Next, the clips are input into the model, which outputs *bounding boxes* or *tubelets*. Finally, these outputs are linked using IoU to create tubes. (b) The proposed method directly outputs *tubes* of the original video.

The second issue involves splitting a video into video clips. Regardless of the video's context, clips that are trimmed to a predetermined length may not be suitable for action recognition. For example, a clip that only captures the first half of a "jump" action might actually show the same movements as a "crouch" action, but still needs to be identified as the "jump" action that will follow. Recognizing actions from clips with incomplete action might lead to poor performance and may also make training more difficult.

In this paper, we propose a method in which the model directly outputs tubes of the original video without using IoU-based linking and clip splitting. Although the goal of STAD is to create tubes, much attention has not been paid to models that directly output tubes. The proposed method eliminates post-processing because the model directly outputs tubes at the inference stage. Instead, the proposed method links the same person using query features. Specifically, inspired by the recent success of DETR (Carion et al., 2020), we adopt an approach that applies query-based detection for each frame and matches queries to find the same person across frames. Our approach is illustrated in Figure 2. By linking with queries, we eliminate the need for IoU-based linking, allowing the detection of actions even with large displacements. Furthermore, by predicting action classes using a sequence of the queries obtained from matching, our method facilitates inputs of variable lengths for action recognition of the entire video.

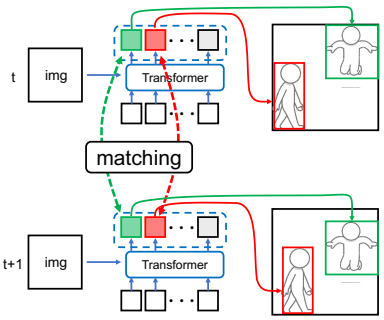

Figure 2: The proposed approach. Linking is performed by matching queries assigned to the same person, eliminating the need for the IoU-based linking.

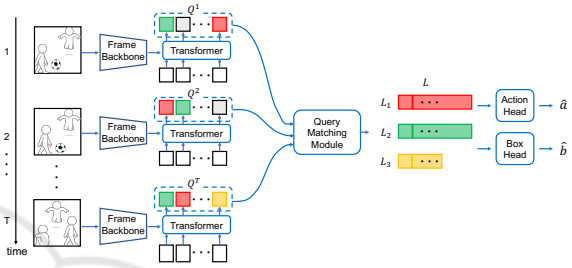

Figure 3: Overview of the proposed method. First, the frame features are obtained using the frame backbone. Next, the frame features and queries interact in a transformer. Then, the proposed Query Matching Module (QMM) matches the queries responsible for the same person in different frames. Finally, the output of QMM is classified by the action head to predict actions and bounding boxes in each frame.

2 METHOD

In this section, we describe the proposed method that directly outputs the tube without using the IoU in the inference stage. Section 2.1 provides an overview and presents our approach for linking the same person using queries. Section 2.2 explains the details of the proposed Query Matching Module (QMM), which is responsible for matching queries to the same person in different frames and construct tubes. Section 2.3 describes the details of the action head, to predict the actions of the tube.

2.1 Overview

First, we explain the approach of linking the same person using queries. DETR (Carion et al., 2020) introduced the concept of queries and performed object detection as a bipartite matching between queries and ground-truth bounding boxes. In other words, one query is responsible for one object. The proposed method learns to match the DETR's object queries re-

sponsible for the same person between frames, which solves the drawbacks of IoU-based linking. Then, by classifying queries of the same person obtained through linking for predicting the action of the tube represented by the queries, we address the drawbacks of the use of clips.

The overview of the proposed method is shown in Figure 3. Our proposed model consists of five components: the frame backbone for extracting frame features, the transformer for interacting the frame features and the queries, the Query Matching Module (QMM) for matching queries that are responsible for the same person, the action head for predicting actions from queries of the tube, and the box head for predicting bounding boxes in the tube.

2.1.1 Steps for Prediction

The following are the steps how the components work.

1. For each frame $x^t \in \mathbb{R}^{3\times H_0\times W_0}$ for the frame index $t = 1,\dots,T$, the frame features $f^t \in \mathbb{R}^{3\times H\times W}$ with dimensions of height H and width W are obtained by the frame backbone. Here, T represents the number of frames in the video, H_0 and W_0 represent the height and width of each frame, respectively.
2. The transformer makes the frame feature f^t interact with N object queries to output a set of queries $Q^t = \{q_i^t\}_{i=1}^N$, where $q_i^t \in \mathbb{R}^d$. This process is performed for each frame t to generate the sets $Q = \{Q^t\}_{t=1}^T$ for all frames $t = 1,\dots,T$.
3. QMM takes the sets Q and outputs $L_j = [q_{j^*}^{t_s}, q_{j^*}^{t_s+1},\dots,q_{j^*}^{t_e}]$. This is a list of queries responsible for person j, starting at frame t_s and ending at frame t_e. Here, $q_{j^*}^t$ is a query $q_i^t \in Q^t$ for some index $i = j^*$ that is predicted to be responsible for the same person j.
4. The action head takes the list L_j of length $t_e - t_s + 1$ to predict a sequence of action scores $\hat{a}_j = [\hat{a}_{j^*}^{t_s}, \hat{a}_{j^*}^{t_s+1},\dots,\hat{a}_{j^*}^{t_e}]$. Here, $\hat{a}_{j^*}^t \in [0,1]^{C+1}$ is the action score predicted at frame t over $C+1$ action classes including "no action". This enables us to predict different actions in different frames of the same person.

2.1.2 Tube Prediction

Based on the prediction $\hat{a}_j$ obtained in the above steps, a tube prediction is generated using the following procedure.

The action score of the tube for person j is the average of the scores of the same action in $\hat{a}_j$. This is formally defined as follows. First, we define the following sets:

$$C_j = \{c \mid c \in \mathrm{top_k}(\hat{a}_{j^*}^t),\ \hat{a}_{j^*}^t \in \hat{a}_j\} \tag{1}$$

$$T_{j,c} = \{t \mid c \in \mathrm{top_k}(\hat{a}_{j^*}^t),\ \hat{a}_{j^*}^t \in \hat{a}_j\}. \tag{2}$$

Here, $\mathrm{top_k}(\hat{a}^t)$ is the operation of extracting the top k elements of the score vector $\hat{a}^t$. C_j is the set of classes with top-k scores in each frame and $T_{j,c}$ is the set of frame indices where the class c is in the top-k.

The scores of the tube for each $c \in C_j$ is then defined as

$$\hat{a}_{j,c} = \begin{cases} \frac{1}{|T_{j,c}|}\sum_{t\in T_{j,c}} \hat{a}_{j^*,c}^t & |T_{j,c}| > \tau_k \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where τ_k is a threshold for filtering out short predictions.

The bounding boxes of the tube for action c is also defined as

$$\hat{b}_{j,c} = \{f_b(q_{j^*}^t) \mid t \in T_{j,c},\ q_{j^*}^t \in L_j\}, \tag{4}$$

where $f_b : \mathbb{R}^d \mapsto [0, H-1]\times[0, W-1]$ is the box head that predicts the bounding box of the query.

The parameters of the pre-trained DETR are fixed and used for the frame backbone, transformer, and box head, while the QMM and action head are trained separately. Next, the details of the QMM and the action head will be explained.

2.2 Query Matching Module

The proposed Query Matching Module (QMM) is trained using metric learning to match queries that are responsible for the same person across frames. Specifically, the query features of the same person are brought closer together, while those of different persons are pushed apart. In the following, we formulate the loss for the training, followed by the inference process.

2.2.1 Training

For training the matching queries of the same person, we do not use all frames of the video. Instead, we use a clip of T frames extracted from the original video.

Given the sets of input queries $Q = \{Q^t\}_{t=1}^T$, first we filter out queries to retain only those classified as "person" with high confidence. This is done by examining the object class score c_i^t

$$c_i^t = f_c(q_i^t), \quad q_i^t \in Q^t, \tag{5}$$

obtained from $f_c : \mathbb{R}^d \mapsto \mathbb{R}^{(C+1)\times[0,1]}$, the head of class prediction of the pre-trained DETR.

Then we select the score of the person class from c_i^t among all $C+1$ classes, and queries are kept if their scores exceed the threshold τ_p; otherwise, they are excluded.

$$Q^{t\prime} = \{q_i^t \mid c_i^t[\text{“person”}] \geq \tau_p\} \tag{6}$$

Next, we find which person is responsible for each query in $Q^{t\prime}$. The prediction of the bounding box $b_i^t = f_b(q_i^t)$ obtained using f_b is compared with the ground truth box g_j^t of person j to compute the IoU. We assign the person ID j to only queries q_i^t that exceed the threshold τ_{iou}, otherwise -1, as follows;

$$y_i^t = \begin{cases} j, & \text{if IoU}(b_i^t, g_j^t) > \tau_{iou} \\ -1, & \text{otherwise.} \end{cases} \tag{7}$$

Next, the person feature encoder f_p is used to transform the query to features $e_i^t = f_p(q_i^t)$, which is suitable for matching. These steps are performed for all frames within the clip, resulting in the set $E = \{e_i^t \mid q_i^t \in Q^{t\prime}\}$ to perform contrastive learning with the following N-Pair Loss (Sohn, 2016);

$$l_i^t(e_i^t) = -\log\left(\frac{\sum_{e_j^u \in E_i^{t+}} \exp(\frac{1}{\tau_t}\text{sim}(e_i^t, e_j^u))}{\sum_{e_k^u \in E - \{e_i^t\}} \exp(\frac{1}{\tau_t}\text{sim}(e_i^t, e_k^u))}\right). \tag{8}$$

Here, $E_i^{t+} = \{e_j^u \mid y_j^u = y_i^t, e_j^u \in E - \{e_i^t\}\}$ is the positive example set, that is, the subset of E where each element e_j^u has the same person ID y_i^t with e_i^t. $\text{sim}(\cdot,\cdot)$ is the cosine similarity and τ_t denotes the temperature. We train the person feature extractor f_p to minimize this contrastive loss $\mathcal{L}$,

$$\mathcal{L} = \sum_{t=1}^{T} \frac{1}{|E|} \sum_{e_i^t \in E} l_i^t(e_i^t). \tag{9}$$

2.2.2 Inference

The goal of QMM during inference is to produce a set of lists $L = \{L_j\}$ whose element L_j contains queries representing the person j, with L initialized as $\emptyset$.

First, as in the training phase, a set $Q^{t\prime} = \{q_i^t\}$ is formed by selecting only queries with high person class scores to focus on those queries responsible for persons. Then, the person feature vectors $e_i^t = f_p(q_i^t)$ are computed for each query, and also $e_j = f_p(L_j[|L_j|-1])$ for the most recent (or last) elements in the lists $L_j \in L$. Lastly, the similarities between e_j and e_i^t are calculated to verify whether the current query q_i^t represents the same person as L_j. If similarity exceeds the threshold τ_s, the person represented by q_i^t is considered the same as L_j, and q_i^t is added to L_j.

If $L = \emptyset$, it is considered that no persons have appeared in the video before and lists $L_i = [q_i^t]$ are added to L for each $q_i^t \in Q^{t\prime}$. If the similarity is below the threshold, $[q_i^t]$ is considered a newly detected person in the video, not assigned to any $L_j \in L$, and added to L as a new list.

This process is iterated from $t = 1$ to T, and lists in L with a minimum length of τ_k' are used as the final output of QMM.

2.3 Action Head

The action head takes $L_j = [q_{j^*}^{t_s}, \ldots, q_{j^*}^{t_e}]$, the QMM output, and predicts the actions as $\hat{a}_j = [\hat{a}_{j^*}^{t_s}, \hat{a}_{j^*}^{t_s+1}, \ldots \hat{a}_{j^*}^{t_e}]$. L_j represents the same person in successive frames, regardless of the presence, absence, or changes in actions. Therefore, the action head predicts actions $\hat{a}_{j^*}^t \in \mathbb{R}^{C+1}$, including “no action,” for every frame $t = t_s, \ldots, t_e$ in L_j, rather than a single action for the entire L_j.

Since the length of L_j varies for different j, a transformer is used as the action head with time encoding. This encoding adds (or concatenates) time information to encode each q^t in L_j with its time t relative to t_s. The action head also has cross-attention from the frames. A pre-trained action recognition model is applied to the frames to obtain a global feature, which is then used as a query in the attention within the transformer of the action head.

Note that the action head classifies the elements in L_j, which are the original DETR query q_i^t rather than the person feature $f_p(q_i^t)$. This choice stems from the differing goals of metric learning and action classification. Metric learning aims to keep output features distinct for different individuals, even when performing the same action. In contrast, action classification needs similar features for the same action, regardless of the person performing it. Therefore, encoded features $f_p(q_i^t)$ are employed for metric learning, while DETR queries q_i^t are used for action classification.

The loss for actions is a standard cross entropy;

$$\mathcal{L} = \sum_{L_j \in L} \sum_{t=t_s}^{t_e} L_{CE}(\hat{a}_{j^*}^t, a_{j^*}^t). \tag{10}$$

Here, action labels of each frame t is

$$a_{j^*}^t = \begin{cases} \text{“no action”}, & \text{if } y_{j^*}^t = -1 \\ c_j^t, & \text{otherwise,} \end{cases} \tag{11}$$

and c_j^t is the action class for person j at frame t,

3 EXPERIMENTAL RESULTS

The proposed method is evaluated using datasets commonly used for evaluating STAD.

3.1 Settings

3.1.1 Datasets

JHMDB21 (Jhuang et al., 2013) includs movies and YouTube videos, consisting of 928 videos with 15 to 40 frames each. There are 21 action classes, and each video contains exactly one ground truth tube of the same length as the video.

UCF101-24 (Singh et al., 2017) consists of 3207 videos with clips ranging from approximately 3 to 10 seconds. There are 24 action classes, and unlike JHMDB, one video can contain any number of ground truth tubes of any length. However, all tubes within a single video belong to the same action class.

AVA (Gu et al., 2018) comprises 430 15-minute videos collected from movies. It features 80 action class labels, with 60 used for evaluation. Each video contains multiple ground truth tubes of varying lengths, allowing tubes of different action classes to coexist within a single frame. The annotations, provided at 1-second intervals, are insufficient for training. To address this, we interpolate annotations during both training and inference using linear interpolation and object detection. For linear interpolation, when the same person performing the same action is annotated in both the starting and ending frames of the target interval, we linearly interpolate the bounding box coordinates between these two frames. However, linear interpolation has the drawback of large errors in bounding boxes when the person's position changes significantly. Therefore, we apply object detection to each frame using YoLoX (Ge et al., 2021) pre-trained on COCO (Lin et al., 2014). We compare the Intersection over Union (IoU) between bounding boxes obtained by the YoLoX detection and linear interpolation, and use the person ID and action ID of the linear interpolated bounding boxes with the highest IoU with the detected bounding boxes.

3.1.2 Model

For the frame backbone, transformer, box head f_b and f_c, we used DETR (Carion et al., 2020) pre-trained on COCO (Lin et al., 2014) and fixed the parameters. However, we fine-tuned DETR's parameters for the UCF101-24 dataset due to discrepancies between the detection boxes of COCO-trained DETR and the annotation boxes. In QMM, the person feature encoder f_p is a 3-layer MLP, and the action head f_a uses a two-layer transformer encoder. The global feature used in the action head is computed by X3D-XS (Feichtenhofer, 2020), a lightweight CNN-based action recognition model, pretrained on Kinetics (Kay et al., 2017).

3.1.3 Training and Inference Strategies

QMM and action head are trained separately. First, QMM is trained, followed by the training of the action head.

QMM is trained for a clip of eight frames extracted at 4-frame intervals from a random start time in the video for JHMDB and UCF101-24, while for AVA, a randomly selected key frame is used as the starting frame. Each frame is resized while maintaining its aspect ratio so that the longer side becomes 512 pixels, which is the input size of the model. The shorter side is padded with black to create the clip. The training settings are as follows: 20 epochs, batch size of 8, AdamW optimizer with an initial learning rate of 1e-4 (decayed by the factor of 10 at epoch 10). Furthermore, $\tau_{iou} = 0.2, \tau_t = 1$, and $\tau_p = 0.75$ for JHMDB, $\tau_p = 0.5$ for UCF101-24 and AVA.

The action head is trained using the output from QMM inference. During inference, frame-by-frame preprocessing remains the same as in the training phase. For JHMDB and UCF101-24, all frames are used as input. However, for AVA, due to memory constraints, the number of input frames during inference is limited to 16. The parameter settings are as follows: For JHMDB, $\tau_p = 0.9$, $\tau_s = 0.5$, $\tau'_k = 8$, for UCF101-24, $\tau_p = 0.5$, $\tau_s = 0.25$, $\tau'_k = 16$, and for AVA, $\tau_p = 0.5, \tau_s = 0.25, \tau'_k = 8$. The training settings are as follows: 20 epochs, AdamW optimizer with initial learning rate of 1e-3 (decayed by the factor of 10 at epoch 15). When generating tubes, $\tau_k = 8$ is used.

3.1.4 Metrics

For JHMDB and UCF101-24, we adopt video-mAP (v-mAP) as the most commonly used evaluation metric for STAD. This is the class average of average precision based on the spatio-temporal IoU (3D IoU) between predicted tubes and ground truth tubes for each action class. On the other hand, for AVA, we use frame-mAP on annotated key frames as the evaluation metric, as it is the official protocol.

We also evaluated tube detection solely to confirm the effectiveness of the proposed QMM. Specifically, we evaluate using the recall of predicted tubes against the regions of ground truth tubes. We consider a prediction correct if the 3D IoU between the ground truth tube and the predicted tube is above a threshold and

Table 1: Comparison of recall performance between QMM and IoU-based linking. The 3D IoU threshold is set at 0.5 for JHMDB and 0.2 for both UCF101-24 and AVA.

	JHMDB				UCF				AVA			
	All	L	M	S	All	L	M	S	All	L	M	S
IoU $\geq$ 0.25	**92.9**	**75.8**	73.2	**92.3**	82.9	57.4	**55.5**	68.0	87.7	25.8	20.0	81.3
IoU $\geq$ 0.5	91.0	63.6	73.2	91.8	78.6	44.4	49.6	**71.1**	90.9	21.7	16.7	86.9
IoU $\geq$ 0.75	81.0	27.3	53.7	88.7	50.7	4.00	21.2	68.0	**93.0**	14.1	12.9	**92.3**
QMM	91.4	**75.8**	**75.6**	91.2	**83.3**	**60.1**	53.9	68.7	83.9	**37.8**	**23.0**	69.1

incorrect otherwise; then calculate the recall. Since the STAD dataset only annotates people performing actions, we use only recall for evaluation rather than precision, which is affected by other people who are not performing any actions. For the same reason, we use the 3D IoU only at frames where the ground truth tube exists when calculating the recall.

Furthermore, we perform evaluations based on the magnitude of action position changes. We adopt the motion category in MotionAP (Singh et al., 2023), and calculate the recall for ground truth tube regions in each of the Large (L), Medium (L), and Small (S) motion categories.

3.2 Results

3.2.1 Performance of Query Matching by QMM

Table 1 shows a comparison of the proposed QMM versus IoU-based linking. In the table, "All" represents the recall for all ground truth tubes regardless of motion category, while "L," "M," and "S" represent the recall for each motion category. The 3D IoU threshold is set to 0.5 for JHMDB and 0.2 for UCF101-24 and AVA.

The performance of IoU-based linking decreases for categories with larger motions as the threshold increases for both datasets, indicating that using a smaller IoU threshold value could mitigate the drawbacks of IoU-based linking. However, using a small threshold value leads to fundamental problems such as easier linking with false detections and increased possibility of linking failures in situations where people overlap.

In contrast, QMM, compared to using an IoU threshold of 0.75, the proposed QMM performs better than IoU-based linking with an IoU threshold of 0.75. Compared to an IoU threshold of 0.5, it improves in the All, Large, and Medium categories, while performance decreases in the Small category. Compared to an IoU threshold of 0.25, for JHMDB and AVA, performance decreases in the All and Small categories, but is equal to or better in the Large and Medium categories. However, for UCF101-24, performance improvements are observed in the All, Large and Small categories. From these findings, QMM is particularly effective for actions with large position changes.

Table 2: Ablation study on the use of time encoding and global features in the action head. Performance is shown as v-mAP@0.5 for JHMDB, and v-mAP@0.2 for UCF101-24.

time encoding	global feature	JHMDB top1	JHMDB top5	UCF top1	UCF top5
-		28.7	39.0	42.4	49.4
add		31.9	42.1	44.0	50.1
concat		35.7	46.5	45.0	51.1
concat	✓	41.7	53.3	61.0	65.6

3.2.2 Ablation Study of Action Head

Here, we show an ablation study on the action head. Specifically, we compare the presence and absence of time encoding, which adds temporal information to the input, and global features obtained from a pre-trained model. Table 2 shows the results based on v-mAP. Note that top1 and top5 represent the top-k values in Eqs (1) and (2), respectively. The 3D IoU threshold is set to 0.5 for JHMDB and 0.2 for UCF101-24, consistent with the previous experiment. In other words, the performance is shown as v-mAP@0.5 for JHMDB and v-mAP@0.2 for UCF101-24. Experiments on the AVA dataset is ongoing and will be presented in the final version.

Using the time encoding was effective in both datasets. In particular, concatenation showed a more pronounced effect than addition. This suggests that explicit incorporation of temporal information is crucial for effective action recognition.

Next, we can see that incorporating global features from a pre-trained action recognition model significantly improves performance in both datasets. This improvement leads to two key insights. First, global features support the learning of the action head, which needs to be trained on the output queries of the pre-trained DETR. Using these global features for cross-attention in the action head, the head is likely to achieve more effective learning from scratch. Second, using DETR's output, which is trained on object detection tasks, for action recognition has limitations.

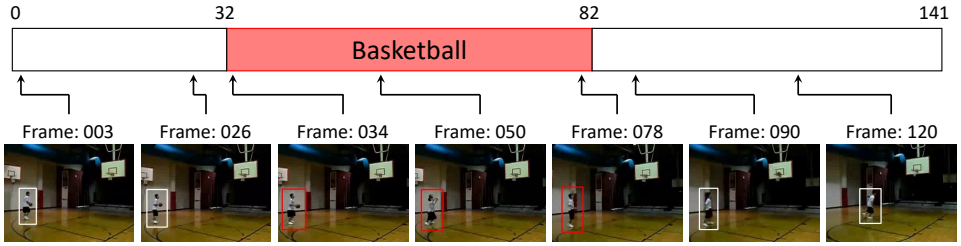

Figure 4: Visualization of predictions by the proposed method. (Top) Temporal annotation intervals. (Bottom) The bounding boxes represent predicted tubes. Boxes are in when the top-1 class of the frame is "no action", and in red when 'Basketball".

In object detection, all people are labeled as "person," regardless of their actions. As a result, DETR's queries may lack action-specific information. However, global features can fill this gap, boosting action recognition performance.

The differences in the top-k value show that top-5 performs better than top-1. This suggests that while the top-1 action in $\hat{a}^t$ may not be the correct action of the tube, it is often the case that the top-5 actions include. Therefore, increasing the top-k value improves the performance of the v-mAP. However, for videos where the start and end of actions are correctly detected using only the top-1 action, as shown in Figure 4, using the top-5 actions may negatively affect performance due to unnecessary actions being also used to predict the action of the tube. Future work includes improvements such as excluding timestamps from $T_{j,c}$ where the predicted action scores are lower than the "no action" class score.

4 CONCLUSIONS

We have proposed a STAD method that directly generates action tubes without relying on post-processing steps, such as IoU-based linking. Instead, our approach utilizes metric learning between DETR queries to link actions across frames. Although the proposed Query Matching Module (QMM) and action head are trained using fixed-length clips, our method offers a flexible inference framework that can handle variable-length video inputs, enhancing its adaptability to diverse videos. In future work, our aim is to extend this approach to more complex tasks, such as spatio-temporal sentence grounding (Yang et al., 2022), and to further refine the architecture and learning strategies to improve the performance of query matching to track the same person across different frames.

ACKNOWLEDGMENTS

This work was supported in part by JSPS KAKENHI Grant Number JP22K12090.

REFERENCES

Ahmed, S. A., Dogra, D. P., Kar, S., Patnaik, R., Lee, S.-C., Choi, H., Nam, G. P., and Kim, I.-J. (2020). Query-based video synopsis for intelligent traffic monitoring applications. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3457–3468.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham. Springer International Publishing.

Chen, L., Tong, Z., Song, Y., Wu, G., and Wang, L. (2023). Efficient video action detection with token dropout and context refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10388–10399.

Chen, S., Sun, P., Xie, E., Ge, C., Wu, J., Ma, L., Shen, J., and Luo, P. (2021). Watch only once: An end-to-end video action detection framework. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8158–8167.

Feichtenhofer, C. (2020). X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430.

Gkioxari, G. and Malik, J. (2015). Finding action tubes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gritsenko, A. A., Xiong, X., Djolonga, J., Dehghani, M., Sun, C., Lucic, M., Schmid, C., and Arnab, A. (2023). End-to-end spatio-temporal action localisation with video transformers. *CoRR*, abs/2304.12160.

Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S.,

Sukthankar, R., Schmid, C., and Malik, J. (2018). Ava: A video dataset of spatio-temporally localized atomic visual actions. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6047–6056.

Jhuang, H., Gall, J., Zuffi, S., Schmid, C., and Black, M. J. (2013). Towards understanding action recognition. In *2013 IEEE International Conference on Computer Vision*, pages 3192–3199.

Kalogeiton, V., Weinzaepfel, P., Ferrari, V., and Schmid, C. (2017). Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A. (2017). The kinetics human action video dataset. *CoRR*, abs/1705.06950.

Köpüklü, O., Wei, X., and Rigoll, G. (2021). You only watch once: A unified cnn architecture for real-time spatiotemporal action localization.

Li, Y., Wang, Z., Wang, L., and Wu, G. (2020). Actions as moving points. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 68–84, Cham. Springer International Publishing.

Lin, T., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. In Fleet, D. J., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Singh, G., Choutas, V., Saha, S., Yu, F., and Van Gool, L. (2023). Spatio-temporal action detection under large motion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6009–6018.

Singh, G., Saha, S., Sapienza, M., Torr, P. H. S., and Cuzzolin, F. (2017). Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Sun, C., Shrivastava, A., Vondrick, C., Murphy, K., Sukthankar, R., and Schmid, C. (2018). Actor-centric relation network. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 335–351, Cham. Springer International Publishing.

Wu, C.-Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., and Girshick, R. (2019). Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, A., Miech, A., Sivic, J., Laptev, I., and Schmid, C. (2022). TubeDETR: Spatio-Temporal Video Grounding with Transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16421–16432, New Orleans, LA, USA. IEEE.

Zhao, J., Zhang, Y., Li, X., Chen, H., Shuai, B., Xu, M., Liu, C., Kundu, K., Xiong, Y., Modolo, D., Marsic, I., Snoek, C. G., and Tighe, J. (2022). Tuber: Tubelet transformer for video action detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13588–13597.