



ActionMixer: Temporal action detection with Optimal Action Segment Assignment and mixers

Jianhua Yang^{a,b}, Ke Wang^a, Lijun Zhao^{a,b}, Zhiqiang Jiang^a, Ruifeng Li^{a,*}

^a State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

^b Wuhu Robot Industry Technology Research Institute, Harbin Institute of Technology, Wuhu 241000, China

ARTICLE INFO

Keywords:

Temporal action detection
Dynamic label assignment
Optimal transport

ABSTRACT

In this paper, we propose a novel method for dynamic label assignment in temporal action detection (TAD) called Optimal Action Segment Assignment (OASA). The OASA method converts label assignment into an optimal transportation problem by computing the cost matrix between predicted temporal action segments and groundtruths. The unit transportation cost between a predicted temporal segment and a groundtruth pair is defined as the weighted summation of action classification loss and temporal localization loss. Additionally, we deploy Adaptive Estimation of Candidate Segment Number (AE-CSN) to adaptively determine the number of positive samples for each groundtruth. After formulation, the label assignment problem is converted to find a global optimal assignment plan by minimizing the cost. Therefore, OASA eliminates the need for manually designed prior parameters, which exist in fixed label assignment methods, and improves the generalization of the algorithm between different datasets. To evaluate OASA, we also introduce a simple anchor-free temporal action detector called ActionMixer. It consists of two components: Temporal Mixer and Channel Mixer. The Temporal Mixer employs depth-wise convolution layers with large kernels to capture temporal information, while the Channel Mixer mixes and extracts features across the channel dimension. Extensive experiments conducted on the THUMOS-14, ActivityNet-1.3, and EPIC-Kitchens-100 datasets show that ActionMixer equipped with OASA achieves state-of-the-art performance, surpassing other advanced temporal action detection methods.

1. Introduction

Temporal action detection (TAD) is a fundamental and important challenge in the field of video understanding. Given a video, a TAD method needs to locate the start time and end time of each possible action instance in the video and classify each action. It plays an important role in intelligent security, video editing, somatosensory games, etc.

With the development of deep learning technology, TAD has achieved great success (Li, Cao, & Ye, 2023; Shou, Chan, Zareian, Miyazawa, & Chang, 2017; Sun, Song, Wu, Jia, & Luo, 2021; Zhou, Wang, Li, & Kung, 2020). Advanced temporal action detection methods can be divided into two-stage methods (Chao et al., 2018; Xu, Das, & Saenko, 2017) and one-stage methods (Lin et al., 2021; Lin, Zhao and Shou, 2017; Long et al., 2019; Yang, Peng, Zhang, Fu, & Han, 2020). A two-stage method first generates temporal proposals with multiple anchor sizes (a.k.a sliding window) (Chao et al., 2018; Gao, Yang, Chen, Sun, & Nevatia, 2017; Xu et al., 2017), temporal actionness grouping (Zhao et al., 2017) or frame-wise boundary detection (Gong,

Zheng, & Mu, 2020; Lin, Zhao, Su, Wang, & Yang, 2018), and then uses a sub-network to recognize each proposal, while a one-stage method can efficiently output all action segments and action categories in a video end-to-end, thereby receiving extensive attention.

However, advanced one-stage methods suffer from a drawback known as **fixed label assignment**, which is illustrated in Fig. 1(a). For anchor-based methods (Buch, Escorcia, Ghanem, Fei-Fei, & Niebles, 2019; Lin, Zhao et al., 2017), a set of anchor sizes is predefined to capture action segments of different durations. The groundtruth is then assigned to different pyramid levels based on the anchor size. Similarly, anchor-free methods (Lin et al., 2021; Yang et al., 2020) predefine a set of scale ranges of interest, such as $[2^{i-1}, 2^i]$, for label assignment, where i is the level of pyramid feature. However, these priors set on one dataset may not be suitable for another dataset, necessitating redesign which may limit the generalization of the algorithm. In summary, *there is a need to study how to design an efficient dynamic label assignment for one-stage TAD.*

* Corresponding author.

E-mail addresses: 19B908049@stu.hit.edu.cn (J. Yang), wangke@hit.edu.cn (K. Wang), zhaolj@hit.edu.cn (L. Zhao), 22S008043@stu.hit.edu.cn (Z. Jiang), lrf100@hit.edu.cn (R. Li).

<https://doi.org/10.1016/j.eswa.2023.121330>

Received 6 December 2022; Received in revised form 31 July 2023; Accepted 24 August 2023

Available online 9 September 2023

0957-4174/© 2023 Elsevier Ltd. All rights reserved.

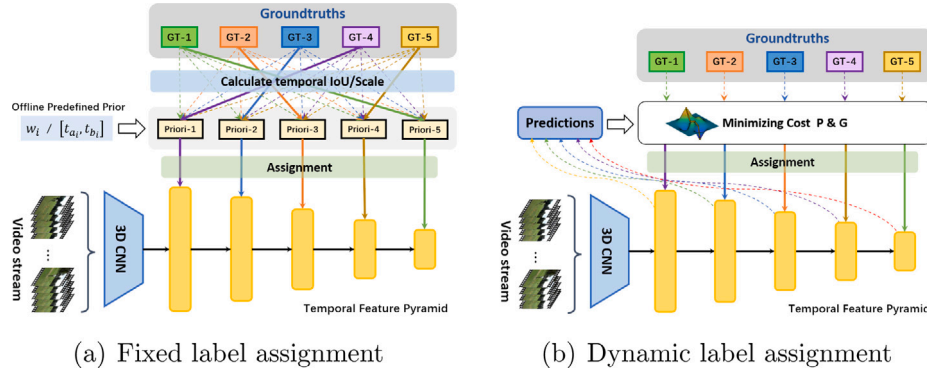


Fig. 1. Overview of label assignment. The **left figure** shows the **fixed label assignment**. Based on the type of priors (e.g. anchor size w_l or scale range of interest $[t_a, t_b]$), we first compute a metric (e.g. tIoU or scale of gt) between a groundtruth and all priors. Then, the predictions marked as positive samples are filtered according to pre-defined criteria (e.g. tIoU threshold or scale range). Positive samples will learn this groundtruth. These priors are often carefully defined offline, affecting the quality of positive samples. The **right figure** shows the **dynamic label assignment**. The cost between a groundtruth and all predictions is directly calculated. Then the optimal label assignment can be obtained by minimizing the cost. Any priors like anchor size or scale range of interest are not required for dynamic label assignment.

In this paper, we propose a novel dynamic label assignment, **Optimal Action Segment Assignment (OASA)** for temporal action detection. OASA converts label assignment to an optimal transport problem. In a given training iteration, OASA first calculates the cost between all predictions and groundtruths. The Adaptive Estimation of Candidate Segment Number (AE-CSN) is designed to determine the number of positive samples for each groundtruth based on the tIoU between each groundtruth and prediction. OASA then uses an iterator to minimize the cost and output the optimal label assignment. Unlike fixed label assignment methods that require priors like anchor size and scale range of interest, OASA does not need them, avoiding the risk of designing different priors for different datasets. Additionally, OASA overcomes the inefficiency of one-to-one matching by deploying AE-CSN. If tIoU is low, it indicates that the quality of the current prediction for this groundtruth is low, and thus the number of positive samples is also low, and vice versa. Compared with fixing the number of positive samples, AE-CSN adaptively determines the most appropriate number for each groundtruth based on the current prediction quality. OASA is expected to provide an effective optimal assignment by globally modeling the association between predictions and groundtruths.

To validate the effectiveness of OASA, we also introduce a new anchor-free and one-stage TAD network, called **ActionMixer**. ActionMixer consists of two components: **Temporal Mixer** and **Channel Mixer**. Temporal Mixer comprises parallel depth-wise convolution layers with large kernels to capture temporal information, and Channel Mixer mixes and extracts features across the channel dimension. The pipeline of ActionMixer is designed to be simple and efficient. With OASA, ActionMixer achieves superior performance on three benchmark datasets, namely THUMOS-14, ActivityNet-1.3, and EPIC-Kitchens-100.

To summarize, the main contributions are as follows:

- A novel dynamic label assignment, called OASA, is proposed in this paper. OASA is expected to give an effective optimal assignment by modeling the association between predictions and groundtruths.
- A new anchor-free and one-stage detector, ActionMixer, is proposed in this paper. It is equipped with Temporal Mixer and Channel Mixer to effectively model the temporal relationships between different action segments.
- ActionMixer, equipped with OASA, achieves superior performance on THUMOS-14, ActivityNet-1.3, and EPIC-Kitchens-100 benchmarks, demonstrating the effectiveness and generalization of OASA in improving the overall performance of ActionMixer.

2. Related work

2.1. Temporal action detection

Current TAD methods can be divided into two-stage methods and one-stage methods. The **two-stage methods** (Chao et al., 2018; Li et al., 2020; Lin, Liu, Li, Ding, & Wen, 2019; Qing et al., 2021; Sridhar et al., 2021; Sun et al., 2021; Tan, Tang, Wang, & Wu, 2021; Xu et al., 2017; Zhu, Tang, Wang, Zheng, & Hua, 2021) comprise a temporal proposal network for localization and a sub-network for classifying each proposal. Early frameworks such as R-C3D (Xu et al., 2017) and TAL-Net (Chao et al., 2018) established the two-stage technical route. Lin et al. (2019) design the BMN with the boundary-matching network for proposal generation, while Qing et al. (2021) study the temporal context information for proposal refinement. Despite their success, these two-stage methods may suffer from being time-consuming and non-end-to-end differentiable.

In contrast to the two-stage method, the **one-stage method** (Lin et al., 2021; Lin, Zhao et al., 2017; Long et al., 2019; Yang et al., 2020) deploy a single network to output predicted action segments with their respective categories in an end-to-end manner. SSAD (Lin, Zhao et al., 2017) is considered one of the earliest one-stage methods, which uses 1D CNN to process video features and a feature pyramid with rich anchor sizes to detect action segments. Lin et al. (2021) present a fully anchor-free method, ASFD, which enhances temporal localization by learning boundary features. These one-stage methods achieve a better trade-off between performance and calculation complexity.

2.2. Fixed label assignment for TAD

For label assignment in TAD, anchor-based methods (Buch, Escorcia, Shen, Ghanem, & Carlos Niebles, 2017; Chao et al., 2018; Lin, Zhao et al., 2017) have to set a fixed tIoU threshold such as 0.5, so that those samples whose tIoU with groundtruth greater than the threshold are assigned as positive samples, and negative samples otherwise. With the priors of anchor sizes, the temporal feature pyramid can be deployed well. As for anchor-free methods (Lin et al., 2021; Yang et al., 2020), they need to design different scale ranges for different levels of feature pyramids such as $[2^{i-1}, 2^i]$, where i is the level of pyramid feature, then the samples that fall in one of the scale ranges are assigned as positive samples at this level. However, the model's generalizability is constrained by the fact that these priors are fixed once they are pre-defined and may not be suitable for other datasets.

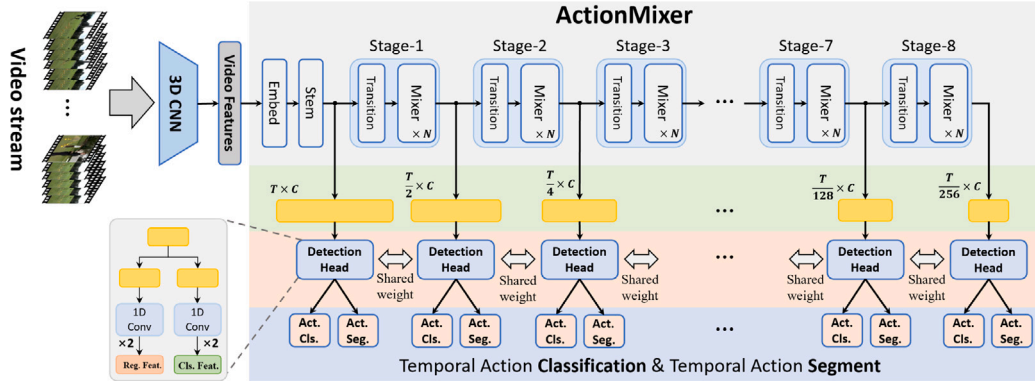


Fig. 2. Structure of ActionMixer: ActionMixer consists of three parts: the embedding layer, the stem layer, and stages that are composed of stacking multiple Mixer modules. ActionMixer collects the outputs from the stem layer and all stages to construct a feature pyramid. The feature of each level is then followed by a detection head with two parallel branches for classification and regression, respectively.

2.3. Dynamic label assignment for TAD

Recently, G-TAD (Xu, Zhao, Rojas, Thabet, & Ghanem, 2020) developed a two-stage TAD approach using a graph neural network. Each groundtruth is assigned only one positive sample during training by G-TAD, which degrades the localization performance. G-TAD assigns the prediction with the highest tIoU to a groundtruth. TADTR (Liu et al., 2021) created a brand-new dynamic label assignment technique called Segment Matcher that is based on the Hungarian algorithm and gets rid of the hassle caused by priors such as anchor sizes and scale ranges of interest. Nevertheless, Segment Matcher likewise assigns each groundtruth with only one positive sample. Although other works use action boundary (Hsieh, Chen, & Liu, 2022; Lin et al., 2019, 2018) detection or temporal actionness grouping (Gong et al., 2020; Zhou et al., 2020) to circumvent these issues, they are two-stage methodologies. In conclusion, designing an effective dynamic label assignment for one-stage TAD approaches remains a challenge.

3. Proposed method

3.1. Preliminary

Given a video V , a 3D CNN first extracts video feature vectors $X = \{x_i\}_{i=1}^T \in \mathbb{R}^{T \times C_{in}}$, where x_i is the feature vector of the video clip at time t , and T is the length of the video. TAD aims to detect all action segments $Y = \{y_i\}_{i=1}^M$. Each action segment $y_i = \{a_i, b_i^s, b_i^e\}$ is defined by its starting time b_i^s , ending time b_i^e and action category a_i . Both b_i^s and b_i^e range from 0 to T , and a_i belongs to $[1, 2, \dots, N_C]$ (N_C is the number of action categories). The duration of each action is equal to the length of the action segment $b_i = b_i^e - b_i^s$.

Given the video feature vectors X , ActionMixer outputs predictions $\hat{Y} = \{\hat{y}_j\}_{j=1}^N$, where each predicted action segment $\hat{y}_j = \{\hat{a}_j, \hat{b}_j^s, \hat{b}_j^e\}$, where $\hat{a}_j \in \mathbb{R}^{N_C}$ is the confidence vector of the action category at the temporal anchor t , \hat{b}_j^s is the distance from the starting time of the action instance to the temporal anchor t , and \hat{b}_j^e is the distance from the ending time to the temporal anchor t . Both \hat{b}_j^s and \hat{b}_j^e are non-negative numbers. Thus, for the temporal anchor t , the starting time and ending time of an action segment are decoded by

$$\hat{a}_j = \arg \max \hat{a}_j, \quad \hat{b}_j^s = j - \hat{b}_j^s, \quad \hat{b}_j^e = j + \hat{b}_j^e \quad (1)$$

During training, label assignment aims to select certain predictions marked as positive samples to learn groundtruths, and therefore plays a crucial role in the model's performance.

3.2. ActionMixer

First of all, we introduce a simple yet strong action detector called **ActionMixer**, which has a one-stage, multi-level, and anchor-free architecture. Its core modules are the Temporal Mixer and Channel Mixer.

Temporal Mixer. Temporal Mixer aims to capture the temporal relationship of actions. One potential solution is the self-attention mechanism (Liu et al., 2021; Wang, Yang, Wu, Yao, & Huang, 2021), but it suffers from a computational complexity of $O(N^2)$. Recent work (Han et al., 2021) on the base model showed that depth-wise convolution with a large kernel is comparable to self-attention. Inspired by this finding, we adopted depth-wise convolution with a 1×9 kernel to design Temporal Mixer. Furthermore, we deployed H parallel depth-wise convolution heads to capture rich features, similar to the multi-head technology used in Transformer (Vaswani et al., 2017). The mathematical formula for the i th head is as shown in Eq. (2).

$$Z_i = \text{Norm}(\text{DWConv}_i(X)) \quad (2)$$

After the H heads, the resulting features $\{Z_i\}_{i=1}^H$ are concatenated along the channel dimension and then processed by another depth-wise convolution with 1×9 kernel followed by the ReLU to generate the output. To prevent gradient disappearance, a residual connection is employed, as shown in Eq. (3).

$$Z = \text{Concatenate}[Z_1, Z_2, \dots, Z_H] \\ Y_T = \text{ReLU}(\text{DWConv}(Z)) + X \quad (3)$$

After applying the Temporal Mixer, the temporal correlation between each action's features is strengthened, leading to improved positioning ability along the timeline.

Channel Mixer. The Channel Mixer is designed to enable features to interact in the channel dimension, which is lacking in the Temporal Mixer due to the separable characteristic of depth-wise convolutions. To address this issue, the Channel Mixer uses a pointwise convolution layer with a 1×1 kernel, followed by LayerNorm and ReLU , to mix the features along the channels. A residual connection is also employed to prevent gradient disappearance, as illustrated in Eq. (4).

$$Z = \text{PTConv}(Y_T) \\ Y_C = \text{ReLU}(\text{Norm}(Z)) + Y_T \quad (4)$$

Finally, the Temporal Mixer cascades a Channel Mixer to construct a Mixer module, as shown in Fig. 3. By combining their respective characteristics, a Mixer module can fully mix and interact with the features of each action in the temporal dimension.

ActionMixer ActionMixer is constructed by simply cascading multiple Mixer modules, as shown in Fig. 2. The architecture starts with an embedding layer that utilizes two ordinary 1D convolution layers

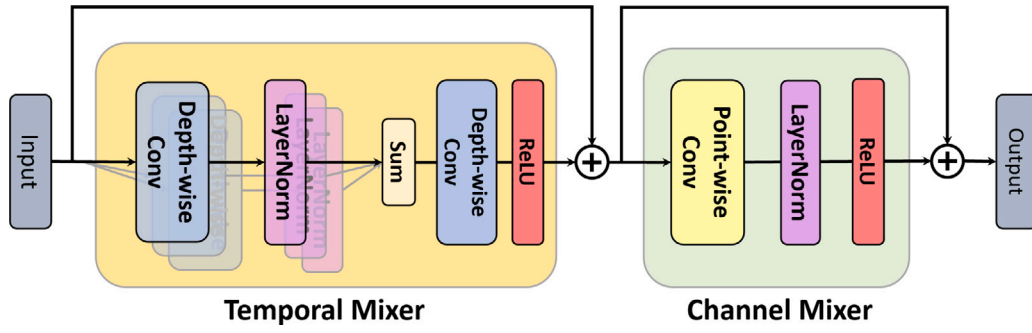


Fig. 3. Structure of Mixer. A Mixer consists of a Temporal Mixer and a Channel Mixer.

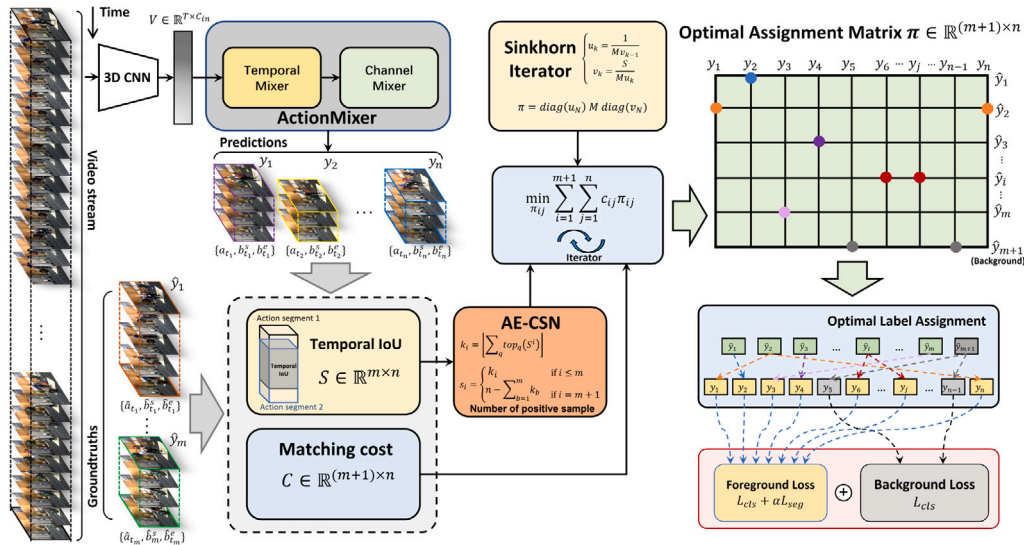


Fig. 4. Overview of Optimal Action Segment Assignment: During training, the video features are extracted by a 3D CNN, and then the proposed ActionMixer processes the features to obtain predictions. Next, the cost matrix between predictions and ground truths is calculated, and the Adaptive Estimation of Candidate Segment Number is used to dynamically determine how many predictions should be assigned to each ground truth y_i . The optimal solution is then solved iteratively using related mature algorithms such as the Sinkhorn algorithm. Finally, the optimal label assignment plan can be decoded from the solution. After the assignment, foreground loss and background loss can be calculated to optimize ActionMixer.

to extract high-level features from video feature vectors. Then, two Mixers are cascaded to build a stem layer. Next, for each stage, N Mixers are cascaded followed by a transition layer, which is a normal 1D convolution layer with a 2 output stride, to build a hierarchical feature pyramid. The mathematical process for each stage is illustrated in Eq. (5).

$$\begin{aligned} Y^l &= \text{Norm}(\text{Trans}(Y^{l-1})), \quad l = 1, 2, \dots, L \\ Y^l &= \text{Mixer}(Y^l), \quad l = 1, 2, \dots, L \end{aligned} \quad (5)$$

where, $Y^l \in \mathbb{R}^{T_l \times C}$ is the output of the l th stage.

To build a feature pyramid, the outputs $Y = \{Y^l\}_{l=1}^L$ from the stem layer and all stages are collected. Each level feature Y^l is then processed by a detection head, consisting of four convolution layers. The first two layers extract features related to action categories, while the second two layers extract features related to temporal localizations. Finally, two additional 1D convolution layers are used to output action segments and action categories, respectively. The parameters of the detection head are shared across all levels.

For inference, the post-processing of ActionMixer, which involves thresholding and non-maximum suppression, follows the same procedure as other anchor-free TAD methods such as those described in Lin et al. (2021) and Yang et al. (2020).

3.3. Optimal Action Segment Assignment

In this section, we provide a detailed introduction to Optimal Action Segment Assignment (OASA). OASA is designed to determine the optimal assignment between all predictions and ground truths. The optimal assignment should minimize the foreground cost $C^{fg} \in \mathbb{R}^{m \times n}$ between predicted segments $\{\hat{y}_j\}_{j=1}^n$ and ground truths $\{y_i\}_{i=1}^m$, and background cost $C^{bg} \in \mathbb{R}^{1 \times n}$, as the following formula:

$$\begin{aligned} c_{ij}^{fg}(\hat{y}_i, y_j) &= L_{cls}(\hat{\theta}_j, a_i) + \gamma L_{seg}(\hat{l}_j^s, \hat{r}_j^e, b_i^s, b_i^e) \\ c_j^{bg}(y_j, \emptyset) &= L_{cls}(\hat{\theta}_j, \emptyset) \end{aligned} \quad (6)$$

where the classification cost L_{cls} is the focal loss (Lin, Goyal, Girshick, He and Dollár, 2017), regression cost L_{seg} is the GIoU loss (Rezatofighi et al., 2019), and γ is the cost balance factor, set to 1.5. We combine foreground cost and background cost to get the full cost matrix $C \in \mathbb{R}^{(m+1) \times n}$. Therefore, the optimal solution can be obtained by minimizing the following cost:

$$\pi = \arg \min \sum_{i=1}^{m+1} \sum_{j=1}^n C(\hat{y}_i, y_j) \quad (7)$$

where $\pi \in \mathbb{R}^{(m+1) \times n}$ is the optimal solution, and π_{ij} represents the assignment between groundtruth y_i (including background labels) and predicted segment \hat{y}_j .

It can be observed that the essence of this problem is an **optimal transportation** problem. Therefore, we view each groundtruth y_i as a supplier who provides p_i units of positive samples, and each predicted segment \hat{y}_j as a demander who requires d_j unit of the label. Since a predicted segment can only learn one groundtruth, d_j is equal to 1. Combined with the definition of label assignment in TAD, p_i is equal to k_i . Finally, we can reformulate the objective function defined in Eq. (7).

$$\begin{aligned} \min_{\pi_{ij}} \quad & \sum_{i=1}^{m+1} \sum_{j=1}^n c_{ij} \pi_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^{m+1} \pi_{ij} = 1, \quad j = 1, 2, \dots, n \\ & \sum_{j=1}^n \pi_{ij} = p_i, \quad i = 1, 2, \dots, m+1 \\ & \pi_{ij} \geq 0, \quad i = 1, 2, \dots, m+1, j = 1, 2, \dots, n \end{aligned} \quad (8)$$

where c_{ij} is the cost between the j th predicted action segment and the i th groundtruth, defined by Eq. (6), π_{ij} is the element of $\pi \in \mathbb{R}^{(m+1) \times n}$ and p_i is equal to k_i if $i \leq m$ and $n - \sum_{b=1}^m p_b$ otherwise. The goal is to minimize the total cost, so as to obtain the optimal matching scheme π .

Algorithm 1 Optimal Action Segment Assignment.

```

1: Input:
2:   X video feature vectors
3:    $\hat{Y} = \{\hat{Y}_{cls}, \hat{Y}_{seg}\}$  Groundtruths
4:    $\gamma$  cost balance factor
5:    $\epsilon = 0.1$  hyperparameters of SinkhornIter
6:    $N = 50$  the number of iteration of SinkhornIter
7: Output:  $\pi \in \mathbb{R}^{(m+1) \times n}$  optimal label assignment
8:    $Y = \{Y_{cls}, Y_{seg}\} \leftarrow \text{ActionMixer}(X)$ 
9:    $n \leftarrow |Y|, m \leftarrow |\hat{Y}|$ 
10:   $S \leftarrow \text{Initialized to Zeros } \mathbb{O} \in \mathbb{R}^{m \times n}$ 
11:  for  $i = 1, \dots, m$  :
12:    for  $j = 1, \dots, n$  :
13:       $S^{ij} \leftarrow tIoU(Y_{seg}^j, \hat{Y}_{seg}^i)$ 
14:  # AE-CSN
15:  for  $i = 1, \dots, m$  :
16:     $p_i = k_i \leftarrow \text{sum}(\text{top}_q(S^i))$ 
17:  # background supplier
18:   $p_{m+1} = n - \sum_{i=1}^m p_i$ 
19:   $C^{fg} \leftarrow \text{Initialized to Zeros } \mathbb{O} \in \mathbb{R}^{m \times n}$ 
20:   $C^{bg} \leftarrow \text{Initialized to Zeros } \mathbb{O} \in \mathbb{R}^{1 \times n}$ 
21:  for  $i = 1, \dots, m$  :
22:    for  $j = 1, \dots, n$  :
23:       $c_{cls}^{ij} = L_{cls}(\hat{Y}_{cls}^j, Y_{cls}^i)$ 
24:       $c_{seg}^{ij} = L_{seg}(\hat{Y}_{seg}^j, Y_{seg}^i)$ 
25:       $c_{ij}^{fg} = c_{cls}^{ij} + \gamma c_{seg}^{ij}$ 
26:  # background cost
27:  for  $j = 1, \dots, n$  :
28:     $c_j^{bg} = L_{seg}(\hat{Y}_{seg}^j, \emptyset)$ 
29:  final cost  $C = \text{Cat}([C^{fg}, C^{bg}], \text{dim} = 0)$ 
30:   $u_0, v_0 \leftarrow \text{Initialized to Ones } \mathbb{I} \in \mathbb{R}^{m+1} \text{ and } \mathbb{R}^n$ 
31:  # SinkhornIter
32:   $M = \exp(-C/\epsilon)$ 
33:  for  $k = 1, \dots, N$  :
34:     $u_k = \frac{1}{M v_{k-1}}, v_k = \frac{P}{M u_k}$ 
35:  optimal solution  $\pi = \text{diag}(u_N) M \text{diag}(v_N)$ 
36:  return  $\pi$ 

```

Now that the label assignment has been transformed into an optimal transportation problem as shown in Eq. (8), we can leverage relevant

mathematical tools to minimize the cost, so as to find the optimal solution. Sinkhorn (Cuturi, 2013) is an efficient and effective algorithm for solving optimal transportation problems, so we use it to minimize the cost defined in Eq. (8). Following the requirements of the Sinkhorn algorithm, we first calculate the matrix $M \in \mathbb{R}^{(m+1) \times n} = \exp(-\frac{C}{\epsilon})$, where the C is the cost matrix. Then we start iterating using the following formulas,

$$u_k = \frac{1}{M v_{k-1}}, v_k = \frac{P}{M u_k}, k = 1, \dots, N \quad (9)$$

where the elements of $u_0 \in \mathbb{R}^{m+1}$ and $v_0 \in \mathbb{R}^n$ are all initialized to 1, and $P \in \mathbb{R}^{(m+1)}$ is the supplier vector. Once the iteration is over, the optimal solution π can be decoded by the Eq. (10).

$$\pi = \text{diag}(u_N) M \text{diag}(v_N) \quad (10)$$

However, the element p_i of P is still unknown, that is, k_i is yet unknown. To address this issue, we use an adaptive estimation method called **Adaptive Estimation of Candidate Segment Number** (AE-CSN), which is inspired by OTA (Ge, Liu, Li, Yoshie, & Sun, 2021) in object detection. Specifically, we first calculate the tIoU between each groundtruth y_i and all predicted segments $\{\hat{y}_j\}_{j=1}^n$. Then, we sum the top q tIoU values and round it up to obtain the value of k_i . Here, we set q empirically to 20.

Theoretically, Eq. (7) represents global optimization. However, a global search may introduce low-quality positive samples. To address this issue, we use center sampling strategy (Lin et al., 2021; Yang et al., 2020) with a radius of r , where only the r temporal anchor points in the central neighborhood of the temporal action segments are considered as high-quality positive sample candidates for calculating the foreground cost.

Once an optimal solution is obtained for label assignment, k_i positive samples \hat{y}_j from n predictions are expected to be assigned to each groundtruth y_i , and the remaining $n - \sum_{i=1}^m k_i$ predictions are marked as negative samples. Note that no fixed parameters or priors like the tIoU threshold or scale range of interest are required, so OASA is a fully dynamic label assignment. The overview of OASA is shown in Fig. 4. Pseudocode is shown in Algorithm 1. For the Sinkhorn algorithm, we empirically set ϵ and N to 0.1 and 50, respectively. We next use SinkhornIter to denote the Sinkhorn iterative algorithm.

3.4. Loss function

The loss function is defined by the following formula:

$$\begin{aligned} L(\hat{a}_j, \hat{t}_j^s, \hat{r}_j^e) = & \frac{1}{N_{pos}} \sum_{j=1}^n L_{cls}(\hat{a}_j, a_j) + \\ & \alpha \frac{1}{N_{pos}} \sum_{j=1}^n \mathbb{I}_{a_j > 0} (L_{seg}(\hat{t}_j^s, \hat{r}_j^e, b_j^s, b_j^e)) \end{aligned} \quad (11)$$

where L_{cls} is the focal loss, L_{seg} is the $GIoU$ loss and N_{pos} is the number of positive samples. L_{cls} is calculated over all locations on the feature map. $\mathbb{I}_{a_j > 0}$ is the indicator function, being 1 if $a_j > 0$ and 0 otherwise, hence L_{seg} is only calculated over positive samples. The α is a loss balance factor and is set to 1.0.

4. Experiments

4.1. Datasets

THUMOS-14 (Idrees et al., 2017). The THUMOS-14 dataset contains 20 action categories collected from over 24 h of videos. Following previous works (Chao et al., 2018; Zhao et al., 2020), the validation set with 200 untrimmed videos is used to train the model, and the test set with 213 untrimmed videos is used for evaluation. I3D (Carreira & Zisserman, 2017) pretrained on Kinetics is used to extract the video features on THUMOS-14. We report the mean Average Precision (mAP)

Table 1
Ablation study on the effectiveness of kernel size in Temporal Mixer.

Kernel size	mAP@tIoU(%)					
	0.3	0.4	0.5	0.6	0.7	Avg
3	72.0	68.7	61.9	51.6	36.5	58.2
5	72.5	69.2	62.0	52.2	38.0	58.8
7	73.1	70.2	63.4	53.9	39.5	59.8
9	73.9	70.8	64.4	53.4	40.0	60.5
11	73.8	70.3	63.8	53.1	38.8	59.9

on the test set at multiple tIoU thresholds {0.3, 0.4, 0.5, 0.6, 0.7} as well as the average mAP under the threshold [0.3 : 0.1 : 0.7].

ActivityNet-1.3 (Caba Heilbron, Escorcia, Ghanem, & Carlos Niebles, 2015). The ActivityNet-1.3 dataset is a large-scale action dataset, containing about 20,000 videos with 200 action categories. Following previous works (Chao et al., 2018; Lin et al., 2019, 2018), the training set with 10,024 videos is used to train the model, and the validation set with 4926 videos is used for evaluation. TSP (Alwassel, Giancola, & Ghanem, 2021) is used to extract feature vectors from the input video. We report the mAP on the validation set at multiple tIoU thresholds {0.5, 0.75, 0.95} as well as the mAP under the threshold [0.5 : 0.05 : 0.95].

EPIC-Kitchens-100 (Damen et al., 2022). EPIC-Kitchens-100 is the latest and largest dataset in egocentric vision. It contains 700 variable-length videos with 90,000 actions, collected from 100 h. This newest dataset is very challenging. It is defined as the combination of a verb (action) and a noun (object). Following the official practice (Damen et al., 2022), we report mAP at tIoU threshold {0.1, 0.2, 0.3, 0.4, 0.5} as well as the mAP under the threshold [0.1 : 0.1 : 0.5] on the validation set. In experiments, SlowFast network (Feichtenhofer, Fan, Malik, & He, 2019) pretrained on EPIC-Kitchens is used to extract feature vectors from the input video.

4.2. Implementation details

For both THUMOS-14 and ActivityNet-1.3, we use the AdamW optimizer with a weight decay of 0.05 and an initial learning rate of 0.0001. The learning rate schedule is set as a cosine annealing schedule. To stabilize the early training phase, we use a warm-up strategy for the first 5 epochs.

For THUMOS-14, we train ActionMixer for a total of 50 epochs (excluding the initial warm-up phase) with a batch size of 2. For ActivityNet-1.3, we train for 10 epochs with a batch size of 16. On EPIC-Kitchens-100, the initial learning rate is set to 0.001, and we train for a total of 30 epochs. The remaining training configurations are consistent across all datasets.

4.3. Ablation studies of ActionMixer model design

Before evaluating OASA, we design ActionMixer with a focus on the *kernel size*, *multi head*, *depth*, as described in Section 3.2. For OASA, we temporarily set the center sampling radius r to 2.

Kernel size in Temporal Mixer. First, we conduct an ablation study on the kernel size of ActionMixer, where the number of heads in the Temporal Mixer and the depth of each stage are both set to 1, and the maximum level of the feature pyramid is set to 6. Only the kernel size is set as a control variable and the experimental results are reported on the THUMOS-14 test set. As shown in Table 1, ActionMixer achieves the best performance with a large kernel size of 9, indicating that larger kernels are better at capturing the temporal context information of actions. Therefore, in subsequent experiments, we set the kernel size in Temporal Mixer to 9.

Depth of Mixer per stage. Then, we investigate the effect of the number of Mixers in each stage, where N Mixers are set to extract features between every two scales. We train ActionMixer with different

Table 2
Ablation study on the effectiveness of Mixer's depth.

Mixer depth N	mAP@tIoU(%)					
	0.3	0.4	0.5	0.6	0.7	Avg
1	73.9	70.8	64.4	53.4	40.0	60.5
2	74.2	70.5	64.3	55.0	40.1	60.7
3	73.5	70.3	64.6	53.3	39.5	60.3
4	73.4	70.1	63.2	54.0	40.0	60.3
5	73.7	70.4	64.0	54.2	39.0	60.3

Table 3
Ablation study on the effectiveness of the number of heads in Temporal Mixer. 1* indicates that the residual connection is removed.

Head H	mAP@tIoU(%)					
	0.3	0.4	0.5	0.6	0.7	Avg
1*	73.2	69.3	63.5	53.9	38.6	59.7
1	74.2	70.5	64.3	55.0	40.1	60.7
2	74.4	70.7	64.3	55.3	40.3	60.9
3	73.7	70.1	63.5	53.5	39.9	60.1
4	74.1	70.6	64.0	54.2	39.3	60.4
5	74.2	70.5	63.4	53.7	39.2	60.2

Table 4
Ablation study on the effectiveness of feature pyramid.

Level	mAP@tIoU(%)					
	0.3	0.4	0.5	0.6	0.7	Avg
1	67.2	60.9	51.8	36.5	19.1	47.1
2	71.7	67.9	60.1	47.8	30.8	55.6
3	73.9	69.9	63.7	53.9	39.3	60.1
4	74.0	70.1	63.2	53.3	39.5	60.0
5	74.0	70.1	63.6	54.2	39.4	60.3
6	74.4	70.7	64.3	55.3	40.3	60.9
7	73.7	69.8	63.6	53.7	39.5	60.1
8	74.6	71.1	64.3	54.6	39.7	60.8
9	74.7	71.1	64.7	54.5	40.2	61.0

depths of Mixers on THUMOS-14 and report the experimental results on the test set in Table 2. The best performance is achieved with the $N = 2$ setting, indicating that the depth of each stage should be set to 2 Mixers. Therefore, we set the Mixer's depth per stage to 2 in subsequent experiments.

Number of heads in Temporal Mixer. As shown in Table 3, we first verify the importance of the residual connection. When deploying one head and removing the residual connection (equivalent to depth-wise convolution), the performance is impaired (60.4% mAP vs. 60.7% mAP), and all metrics decrease, indicating the necessity of adopting the residual connection. Additionally, with two parallel heads in the Temporal Mixer, ActionMixer achieves the best performance. More heads impair performance. Therefore, we set the number of heads in each Temporal Mixer to 2 in subsequent experiments.

Feature pyramid. We also search for the optimal number of FPN layers and report the results on the THUMOS-14 test set in Table 4. When the level is set to 1, the feature pyramid loses the advantage of the hierarchical structure, resulting in the worst performance. As the level of the feature pyramid increases, the hierarchical structure becomes more abundant, and the performance gradually improved. When the level is set to 9, the performance reached its best at 61.0% mAP. However, the maximum FPN stride is now 256 with 9 levels, which means that the length of the input video may not be divisible by the max FPN stride if more pyramid levels are added. Therefore, we set the level of the feature pyramid to 9 in subsequent experiments.

Effectiveness of Mixers. To verify the effectiveness and necessity of Mixers, we compare them with normal convolution layers of the same kernel size. We replace all Mixers with normal convolution layers while keeping other network structures and training configurations the same. The comparison results are summarized in Table 5. From the table, we

Table 5

Ablation study on the effectiveness of our Mixers.

Type	mAP@tIoU(%)						Params	Avg
	0.3	0.4	0.5	0.6	0.7	FLOPs		
Normal Conv	73.3	69.8	62.8	52.9	39.0	46.4 B	56.4 M	59.9
Our Mixers	74.7	71.1	64.7	54.5	40.2	27.4 B	16.2 M	61.0

Table 6

Ablation study on the effectiveness of different label assignment (LA).

LA	mAP@tIoU(%)						Avg
	0.3	0.4	0.5	0.6	0.7		
Fixed	72.3	69.5	61.7	52.3	38.6		58.9
OASA	74.7	71.1	64.7	54.5	40.2		61.0

Table 7

Ablation study on the effectiveness of center sampling radius.

r	mAP@tIoU(%)						Avg
	0.3	0.4	0.5	0.6	0.7		
Baseline	72.3	69.5	61.7	52.3	38.6		58.9
0	69.2	65.2	59.2	48.5	34.1		55.3
1	74.2	70.1	63.4	54.1	39.5		60.3
2	74.7	71.1	64.7	54.5	40.2		61.0
4	74.6	70.9	64.6	55.5	41.5		61.4
6	75.1	71.6	64.7	54.7	41.6		61.5
8	74.3	71.4	64.0	54.2	40.9		61.0
10	75.1	72.0	65.3	55.1	41.3		61.8
12	75.1	72.0	65.4	54.8	40.6		61.6
14	74.8	71.0	65.1	54.7	39.9		61.1
10000	74.2	70.6	64.8	54.0	40.1		60.7

Table 8Ablation study on the effectiveness of γ .

γ	mAP@tIoU(%)						Avg
	0.3	0.4	0.5	0.6	0.7		
2.5	74.6	71.4	64.7	54.6	40.7		61.2
2.0	74.9	71.9	64.8	54.2	40.7		61.3
1.5	75.1	72.0	65.3	55.1	41.3		61.8
1.0	74.8	71.4	64.4	55.0	41.2		61.4
0.5	74.7	70.3	63.8	53.4	38.5		60.1

can observe that the detector with Mixers not only performs better but also has fewer FLOPs and parameters, demonstrating its effectiveness. Therefore, the proposed Mixer is not equivalent to normal convolution and is more efficient.


4.4. Ablation studies of Optimal Action Segment Assignment

Effectiveness of OASA. We employ two label assignment strategies: fixed label assignment and OASA. For the fixed label assignment, we set the scale range of interest to $[2^{l-1}, 2^l]$ for the l th level of the feature and also use the center sampling trick. We train ActionMixer on the THUMOS-14 dataset using both label assignment strategies and compare the results on the test set, as shown in Table 6. ActionMixer trained with OASA achieves better performance (58.9% vs. 61.0%). Fine-tuning the scale range might improve the performance, however, this highlights the shortcomings of fixed label assignment.

Cost balance factor γ in transportation cost. As action classification cost and regression cost have different properties, with the former having stronger semantic information and the latter having more positional information, γ plays a crucial role in balancing them. To evaluate its effectiveness, we train ActionMixer with different γ values and summarized the results in Table 8. When we set γ to 1.5, OASA makes ActionMixer achieve the best performance. Therefore, unless otherwise specified, we set γ to 1.5 in subsequent experiments.

Table 9

Ablation study on the effectiveness of adaptive estimation of candidate segment number.

 K	mAP@tIoU(%)						Avg
	0.3	0.4	0.5	0.6	0.7		
1	71.3	67.0	60.4	50.1	36.1		57.0
2	71.9	68.1	60.7	51.6	39.3		58.3
4	73.3	69.1	62.8	53.2	40.4		59.8
6	74.3	71.1	65.1	54.6	41.4		61.3
8	74.3	70.7	64.1	55.0	41.1		61.0
10	74.7	71.1	65.2	55.1	41.5		61.5
12	74.9	71.2	64.7	55.2	42.0		61.6
14	74.8	70.6	65.7	55.0	40.7		61.3
16	74.5	70.9	64.8	55.0	40.9		61.2
18	74.2	70.5	64.1	54.6	41.1		60.9
Adaptive K	75.1	72.0	65.3	55.1	41.3		61.8

Center sampling. To verify the benefit of the center sampling strategy, we train ActionMixer with different center sampling radius (r). Table 7 shows the experimental results, where Baseline indicates the deployment of fixed label assignment. From the table, we observed that ActionMixer performs the worst when r is set to 0, even worse than Baseline. This is because setting r to 0 meant that only the center point of the groundtruth is considered as the candidate positive sample, resulting in the loss of too many high-quality positive samples. As r increases, the number of positive sample candidates for groundtruths also increases, significantly improving the performance. When r is set to 10, ActionMixer achieves the best performance. Notably, even with a very large r of 10000, which means the center prior almost fails, ActionMixer still achieves better performance than Baseline. Therefore, center sampling is helpful in enhancing performance.

AE-CSN. To study the effect of AE-CSN, we use two methods: setting k_i to a fixed value K and AE-CSN. We train ActionMixer with different K values on the THUMOS-14 dataset and summarize the results in Table 9. From the table, we observe that when K is set to 1 which is equivalent to one-to-one matching, the performance is the worst, proving the inefficiency of one-to-one matching. As the value of K increases, the performance also increases, demonstrating the power of one-to-many matching. However, when the value of K is greater than 10, the performance starts to degrade. This phenomenon indicates that the fixed value of k_i is sub-optimal for OASA.

With AE-CSN, OASA can adaptively adjust the number of predicted segments k_i for each groundtruth y_i based on the tIoU. For the convenience of presentation, we compute the mean of all k_i . From Fig. 5, the overall trend of the mean value of k_i tends to increase with iteration, and the fluctuation during the iteration indicates that AE-CSN adjusts the suitable k_i according to the current prediction results. AE-CSN not only avoids the trouble of manually adjusting but also achieves the best performance, 0.2% mAP better than the best performance achieved with fixed k_i values (61.8% mAP vs. 61.6% mAP).

4.5. Ablation studies of Sinkhorn iterative algorithm

Effectiveness of ϵ . First, we conduct an ablation study on ϵ , which is a hyperparameter of the Sinkhorn iterative algorithm (SinkhornIter) as introduced in Eq. (9). Since it affects the iterative performance of SinkhornIter, it determines the suitability of the optimal assignment matrix $x^* \in \mathbb{R}^{(m+1) \times n}$, thereby affecting the performance of ActionMixer. Table 10 summarizes the results. From the table, we observe that ActionMixer achieves the best performance with $\epsilon = 0.1$. The performance is relatively robust to ϵ in the range from 0.01 to 0.1.

Effectiveness of N . We also study the effect of N , which is the number of iterations of SinkhornIter. N directly determines the final optimal assignment matrix π , thereby affecting the performance of ActionMixer. Table 11 summarizes the results. From the table, we can see that ActionMixer achieves the best performance with $N = 50$.

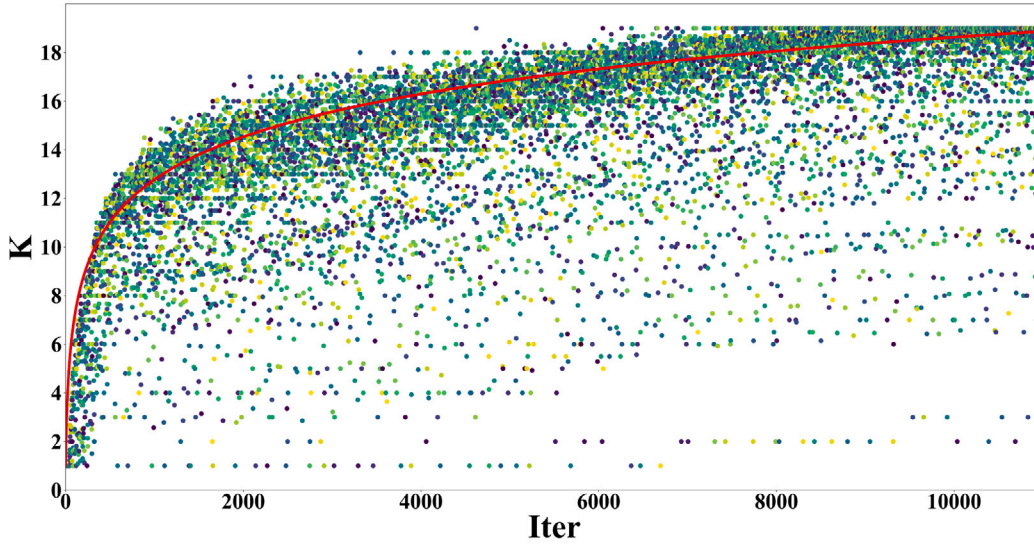


Fig. 5. The distribution of the mean value of k_i during training. It can be observed that as the model is continuously trained, the overall K value shows an upward trend, indicating that the localization performance of the model is continuously improving. Eventually, the K value converges between 18 and 20. The red line is fitted using the RANSAC algorithm.

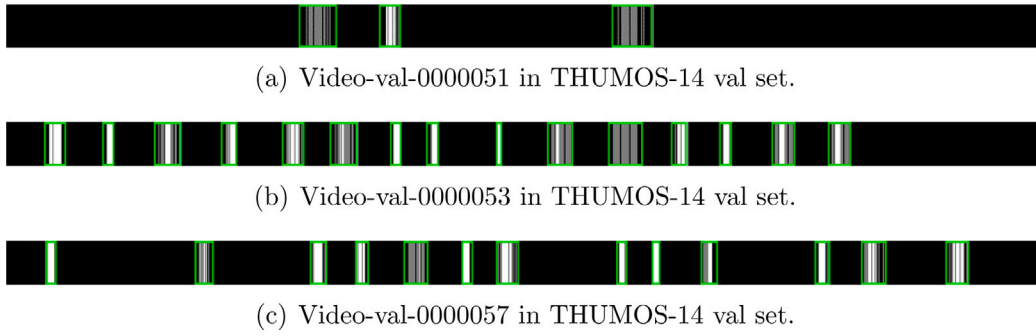


Fig. 6. Qualitative results of positive samples given by OASA. The green bounding boxes represent the action segments and white lines indicate the location of positive samples.

Table 10

Ablation study on the effectiveness of ϵ in SinkhornIter.

ϵ	mAP@tIoU(%)					Avg
	0.3	0.4	0.5	0.6	0.7	
0.001	71.3	67.3	60.6	50.3	36.9	57.3
0.01	74.9	72.0	65.1	55.0	41.2	61.6
0.05	75.0	71.9	65.3	54.8	41.1	61.6
0.1	75.1	72.0	65.3	55.1	41.3	61.8
0.5	73.8	70.1	64.2	53.9	39.8	60.4
1.0	73.4	70.1	63.9	53.2	39.7	60.1

Table 11

Ablation study on the effectiveness of N in SinkhornIter.

N	mAP@tIoU(%)					Avg
	0.3	0.4	0.5	0.6	0.7	
10	74.3	71.0	64.8	54.5	40.9	61.1
50	75.1	72.0	65.3	55.1	41.3	61.8
100	74.8	72.4	64.3	55.0	40.9	61.5
150	74.5	72.0	64.8	53.5	40.0	61.0

Visualization of OASA. To gain an intuitive understanding of the role of OASA, we visualize the label assignments that OASA solves during the training process, which represents the position of positive samples on the time axis. We plot the results as a 2-D image in Fig. 6 for visualization purposes, although the Temporal Action Detection

task is a 1-D detection task in the time dimension. We only need to focus on the horizontal axis because it represents the time dimension. From the figure, we observed that the positions of positive samples given by OASA almost fall within the target action segments, indicating that OASA has learned the correct knowledge about the distribution of positive samples.

4.6. Comparison with state-of-the-art methods

THUMOS-14. The comparison results with state-of-the-art methods on THUMOS-14 are summarized in Table 12. TadTR is an end-to-end TAD method that uses Transformer and Hungarian Matching, but its label assignment belongs to the one-to-one assignment method. From the table, we observe that ActionMixer achieves better performance than TadTR (61.8% mAP vs. 46.6% mAP). This result not only demonstrates the inefficiency of one-to-one assignment but also proves the excellence of OASA. Moreover, this result is consistent with the trend shown in the table, where the one-to-one assignment is indeed inefficient. In addition, compared to AFSD, which is an anchor-free method with fixed label assignment, ActionMixer achieves better performance (61.8% mAP vs. 52.0% mAP), demonstrating the strength of OASA. Therefore, dynamic label assignment is superior to fixed label assignment. Compared to the latest two-stage methods (Tan et al., 2021; Zhao et al., 2021; Zhu et al., 2021), ActionMixer also performs excellently by a large margin. However, other one-stage anchor-free methods, such as ASFD and A2Net, do not show such a significant

Table 12

The mAP results (%) at different tIoU thresholds on THUMOS-14 comparing with state-of-the-art works.

Method	mAP@tIoU(%)					
	0.3	0.4	0.5	0.6	0.7	Avg
R-C3D (Xu et al., 2017)	44.8	35.6	28.9	–	–	–
SSAD (Lin, Zhao et al., 2017)	43.0	35.0	24.6	–	–	–
TAL-Net (Chao et al., 2018)	53.2	48.5	42.8	33.8	20.8	39.8
BSN (Lin et al., 2018)	53.5	45.0	36.9	28.4	20.0	36.8
BMN (Lin et al., 2019)	56.0	47.4	38.8	29.7	20.5	38.5
GTAN (Long et al., 2019)	57.8	47.2	38.8	–	–	–
G-TAD (Xu et al., 2020)	54.5	47.6	40.2	30.8	23.4	39.3
A2Net (Yang et al., 2020)	58.6	54.1	45.5	32.5	17.2	41.6
STA-Net (Li et al., 2020)	53.4	47.5	36.8	–	–	–
BMN-CSA (Sridhar et al., 2021)	64.4	58.0	49.2	38.2	27.8	47.5
ContextLoc (Zhu et al., 2021)	68.3	63.8	54.3	41.8	26.2	50.9
VSGN (Zhao, Thabet, & Ghanem, 2021)	66.7	60.4	52.4	41.0	30.4	50.2
RTD-Net (Tan et al., 2021)	68.3	62.3	51.9	38.8	23.7	49.0
STAN (Sun et al., 2021)	52.8	47.5	39.8	–	–	–
STAN+PGCN (Sun et al., 2021)	67.5	61.0	51.7	–	–	–
TCA-Net (Qing et al., 2021)	60.6	53.2	44.6	36.8	26.7	44.3
TadTR (Liu et al., 2021)	62.4	57.4	49.2	37.8	26.3	46.6
AFSD (Lin et al., 2021)	67.3	62.4	55.5	43.7	31.1	52.0
CPN (Hsieh et al., 2022)	68.2	62.1	54.1	41.5	28.0	50.8
ActionMixer	75.1	72.0	65.3	55.1	41.3	61.8

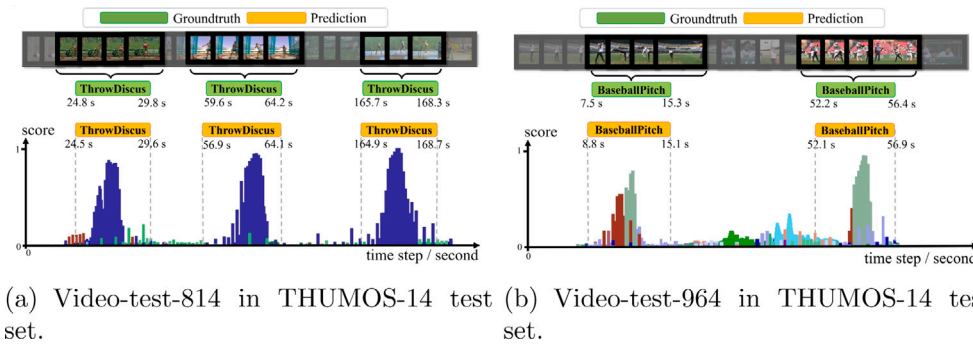


Fig. 7. Qualitative results. For the convenience of presentation, a few video frames are sampled from a video clip to represent the entire action instance. The time boundary for each localized action instance is in seconds.



Fig. 8. Qualitative results on THUMOS-14 test set. “GT” represents the groundtruth, and “PD” represents the predicted action segment.

advantage. In summary, the one-stage anchor-free detector with the support of the proposed OASA can outperform the two-stage model.

For a more intuitive understanding of the proposed work, we present two qualitative results of ActionMixer on the THUMOS-14 test

set in Fig. 7. To represent the entire action instance, we sample a few video frames from a video clip. The time boundary for each localized action instance is shown in seconds. From the figure, we notice that high scores are almost concentrated in the central neighborhood of

Table 13

The mAP results (%) at different tIoU thresholds on ActivityNet-1.3 comparing with state-of-the-art works.

Method	mAP@tIoU(%)			
	0.5	0.75	0.95	Avg
R-C3D (Xu et al., 2017)	28.4	–	–	–
BSN (Lin et al., 2018)	46.5	30.0	8.0	30.0
TAL-Net (Chao et al., 2018)	38.2	18.3	1.3	20.2
BMN (Lin et al., 2019)	50.1	34.8	8.3	33.9
GTAN (Long et al., 2019)	52.6	34.1	8.9	35.5
G-TAD (Xu et al., 2020)	50.4	34.6	9.0	34.1
A2Net (Yang et al., 2020)	43.6	28.7	3.7	27.8
BMN-CSA (Sridhar et al., 2021)	52.4	36.7	5.2	35.4
ContextLoc (Zhu et al., 2021)	56.0	35.2	3.6	34.2
VSGN (Zhao et al., 2021)	52.3	35.2	8.3	34.7
RTD-Net (Tan et al., 2021)	47.2	30.7	8.6	30.8
STAN (Sun et al., 2021)	35.9	21.3	1.7	19.8
TCA-Net (Qing et al., 2021)	52.7	36.7	6.9	35.5
TadTR (Liu et al., 2021)	49.1	32.6	8.5	32.3
AFSD (Lin et al., 2021)	52.4	35.3	6.5	34.4
ActionMixer	53.4	35.8	7.6	35.7

Table 14

The mAP results (%) at different tIoU thresholds on EPIC-Kitchens-100 comparing with baseline.

Task	Method	mAP@tIoU(%)					
		0.1	0.2	0.3	0.4	0.5	Avg
Verb	BMN (Damen et al., 2022)	10.8	9.8	8.4	7.1	5.6	8.4
	ActionMixer	25.4	24.4	22.9	20.5	17.7	22.2
Noun	BMN (Damen et al., 2022)	10.3	8.3	6.2	4.5	3.4	6.5
	ActionMixer	24.1	22.8	21.3	18.9	15.7	20.6

the action segment, which may be due to the center prior used by OASA. More qualitative results are shown in Fig. 8. From the figure, we observe that the proposed method can accurately detect the start time and end time of each action instance, close to the groundtruth. For videos containing multiple action instances, the proposed method also works well.

ActivityNet-1.3. The comparison results with other state-of-the-art works on ActivityNet-1.3 are summarized in Table 13. Thanks to OASA, there is no need to set any hyperparameters for new datasets, demonstrating the generalizability of the ActionMixer. As shown in the table, ActionMixer achieves 35.7% mAP at tIoU threshold [0.5 : 0.05 : 0.95], outperforming other one-stage methods such as AFSD and A2Net. Compared with the TadTR, although the mAP at $tIoU = 0.95$ is slightly lower than TadTR, (7.6% mAP vs. 8.5% mAP), the average mAP at tIoU threshold [0.5 : 0.05 : 0.95] is higher by +3.4% mAP. The advantages demonstrated by ActionMixer on ActivityNet-1.3 once again demonstrate the superiority of the proposed OASA. Compared to the latest two-stage method TCA-Net, ActionMixer also achieves slightly better performance (35.7% mAP vs. 35.5% mAP). Therefore, such experiment results confirm our previous conclusion that the proposed OASA can enable the one-stage anchor-free detectors to deal with the TAD task without the complicated two-stage pipeline.

EPIC-Kitchens-100. In order to further verify the generalization of OASA, we evaluate ActionMixer on the latest released EPIC-Kitchens-100 dataset. Since it is the latest dataset, we can only compare it with the baseline method reported by Damen et al. (2022), which deploys the two-stage method BMN (Lin et al., 2019) as the baseline method. It uses the features from SlowFast to detect action instances. The comparison results are shown in Table 14. For the verb task, ActionMixer achieves 22.2% mAP, much higher than the 8.4% of BMN. For the noun task, ActionMixer also significantly outperforms BMN (20.6% mAP vs. 6.5% mAP). It is worth noting that no priors are designed for this latest dataset, and the configuration of OASA is consistent with the one on the other two datasets. Therefore, such a large performance gap fully proves the excellence of ActionMixer and also demonstrates its excellent generalization performance. This phenomenon is in line with our original intention of designing OASA and our expectations.

5. Conclusion

In this paper, we deal with the temporal action detection task from the perspective of label assignment. We proposed a novel dynamic label assignment method called Optimal Action Segment Assignment (OASA). OASA first calculates the cost matrix between predictions and groundtruths and then uses an iterative algorithm to find the global optimal assignment. Additionally, OASA deploys Adaptive Estimation of Candidate Segment Number (AE-CSN) to determine the number of positive samples for each groundtruth, outperforming the methods that roughly fix the number of positive samples. Compared to other dynamic label assignment methods, OASA takes advantage of one-to-many matching to achieve stronger performance. We also design a new anchor-free and one-stage TAD network, ActionMixer, to verify the effectiveness of OASA. Benefiting from OASA, ActionMixer can effectively utilize the feature pyramid without setting any priors such as anchor size or scale range of interest. Thanks to OASA, ActionMixer can be generalized to different datasets without modification for label assignment. Extensive experiments on popular benchmarks demonstrate the effectiveness of the proposed OASA and ActionMixer. The state-of-the-art performance on multiple datasets, including THUMOS-14, ActivityNet-1.3, and EPIC-Kitchens-100, also demonstrate the generalization of OASA.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data utilized in this paper are publicly available and well-known to researchers in this field. Upon acceptance of the paper, we will promptly release the source code for reproducibility purposes.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62176072).

References

- Alwassel, H., Giancola, S., & Ghanem, B. (2021). TSP: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF international conference on computer vision workshops* (pp. 3173–3183).
- Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., & Niebles, J. C. (2019). End-to-end, single-stream temporal action detection in untrimmed videos. In *Proceedings of the British machine vision conference 2017*. British Machine Vision Association.
- Buch, S., Escorcia, V., Shen, C., Ghanem, B., & Carlos Niebles, J. (2017). Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2911–2920).
- Caba Heilbron, F., Escorcia, V., Ghanem, B., & Carlos Niebles, J. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 961–970).
- Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6299–6308).
- Chao, Y.-W., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., & Sukthankar, R. (2018). Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1130–1139).
- Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26.
- Damen, D., Doughty, H., Farinella, G. M., Furnari, A., Kazakos, E., Ma, J., et al. (2022). Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130(1), 33–55.
- Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6202–6211).
- Gao, J., Yang, Z., Chen, K., Sun, C., & Nevatia, R. (2017). Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision* (pp. 3628–3636).
- Ge, Z., Liu, S., Li, Z., Yoshie, O., & Sun, J. (2021). Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 303–312).
- Gong, G., Zheng, L., & Mu, Y. (2020). Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In *2020 IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). IEEE.
- Han, Q., Fan, Z., Dai, Q., Sun, L., Cheng, M.-M., Liu, J., et al. (2021). On the connection between local attention and dynamic depth-wise convolution. *arXiv preprint arXiv:2106.04263*.
- Hsieh, H.-Y., Chen, D.-J., & Liu, T.-L. (2022). Contextual proposal network for action localization. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 2129–2138).
- Idrees, H., Zamir, A. R., Jiang, Y.-G., Gorban, A., Laptev, I., Sukthankar, R., et al. (2017). The THUMOS challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155, 1–23.
- Li, P., Cao, J., & Ye, X. (2023). Prototype contrastive learning for point-supervised temporal action detection. *Expert Systems with Applications*, 213, Article 118965.
- Li, J., Liu, X., Zhang, W., Zhang, M., Song, J., & Sebe, N. (2020). Spatio-temporal attention networks for action recognition and detection. *IEEE Transactions on Multimedia*, 22(11), 2990–3001.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).
- Lin, T., Liu, X., Li, X., Ding, E., & Wen, S. (2019). Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3889–3898).
- Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., et al. (2021). Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3320–3329).
- Lin, T., Zhao, X., & Shou, Z. (2017). Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on multimedia* (pp. 988–996).
- Lin, T., Zhao, X., Su, H., Wang, C., & Yang, M. (2018). Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–19).
- Liu, X., Wang, Q., Hu, Y., Tang, X., Bai, S., & Bai, X. (2021). End-to-end temporal action detection with transformer. *arXiv preprint arXiv:2106.10271*.
- Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., & Mei, T. (2019). Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 344–353).
- Qing, Z., Su, H., Gan, W., Wang, D., Wu, W., Wang, X., et al. (2021). Temporal context aggregation network for temporal action proposal refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 485–494).
- Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 658–666).
- Shou, Z., Chan, J., Zareian, A., Miyazawa, K., & Chang, S.-F. (2017). Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5734–5743).
- Sridhar, D., Quader, N., Muralidharan, S., Li, Y., Dai, P., & Lu, J. (2021). Class semantics-based attention for action detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 13739–13748).
- Sun, C., Song, H., Wu, X., Jia, Y., & Luo, J. (2021). Exploiting informative video segments for temporal action localization. *IEEE Transactions on Multimedia*.
- Tan, J., Tang, J., Wang, L., & Wu, G. (2021). Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 13526–13535).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Wang, L., Yang, H., Wu, W., Yao, H., & Huang, H. (2021). Temporal action proposal generation with transformers. *arXiv preprint arXiv:2105.12043*.
- Xu, H., Das, A., & Saenko, K. (2017). R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 5783–5792).
- Xu, M., Zhao, C., Rojas, D. S., Thabet, A., & Ghanem, B. (2020). G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10156–10165).
- Yang, L., Peng, H., Zhang, D., Fu, J., & Han, J. (2020). Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29, 8535–8548.
- Zhao, C., Thabet, A. K., & Ghanem, B. (2021). Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 13658–13667).
- Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., & Tian, Q. (2020). Bottom-up temporal action localization with mutual regularization. In *European conference on computer vision* (pp. 539–555). Springer.
- Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., & Lin, D. (2017). Temporal action detection with structured segment networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2914–2923).
- Zhou, Y., Wang, R., Li, H., & Kung, S.-Y. (2020). Temporal action localization using long short-term dependency. *IEEE Transactions on Multimedia*, 23, 4363–4375.
- Zhu, Z., Tang, W., Wang, L., Zheng, N., & Hua, G. (2021). Enriching local and global contexts for temporal action localization. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 13516–13525).

Jianhua Yang is currently working on the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China and Wuhu Robot Industry Technology Research Institute, Harbin Institute of Technology, Wuhu, China. He is the first author and his email address is 19B908049@stu.hit.edu.cn. In this paper, he proposes the key ideas and designs the ActionMixer and OASA, and write this paper.

Ke Wang is currently working on the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China. His email address is wangke@hit.edu.cn. He presents the necessary technical background and proposes a research path for TAD (temporal action detection) problems.

Lijun Zhao is currently working on the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China and Wuhu Robot Industry Technology Research Institute, Harbin Institute of Technology, Wuhu, China. His email address is zhaolj@hit.edu.cn. He provides lots of suggestions for article writing and some computational equipments for the experiments of the paper.

Zhiqiang Jiang is currently working on the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China. His email address is 22S008043@stu.hit.edu.cn. He improves the grammar of the paper, corrects numerous writing errors, and provides valuable suggestions for the article's structure.

Ruifeng Li is currently working on the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China. He is the corresponding author and his email address is lrf100@hit.edu.cn. He provides the experimental platform and necessary computational equipment for the paper.