

Reframing Dense Action Detection (RefDense): A Paradigm Shift in Problem Solving & a Novel Optimization Strategy

Faegheh Sardari¹Armin Mustafa¹Philip J. B. Jackson¹Adrian Hilton¹¹ Centre for Vision, Speech and Signal Processing (CVSSP)

University of Surrey, UK

{f.sardari, armin.mustafa, p.jackson, a.hilton}@surrey.ac.uk

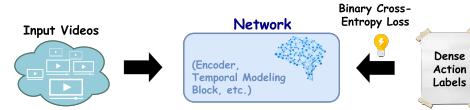
Abstract

*Dense action detection involves detecting multiple co-occurring actions while action classes are often ambiguous and represent overlapping concepts. We argue that handling the dual challenge of temporal and class overlaps is too complex to effectively be tackled by a single network. To address this, we propose to decompose the task of detecting dense ambiguous actions into detecting dense, unambiguous sub-concepts that form the action classes (i.e., action entities and action motions), and assigning these sub-tasks to distinct sub-networks. By isolating these unambiguous concepts, the sub-networks can focus exclusively on resolving a single challenge, dense temporal overlaps. Furthermore, simultaneous actions in a video often exhibit interrelationships, and exploiting these relationships can improve the method performance. However, current dense action detection networks fail to effectively learn these relationships due to their reliance on binary cross-entropy optimization, which treats each class independently. To address this limitation, we propose providing explicit supervision on co-occurring concepts during network optimization through a novel language-guided contrastive learning loss. Our extensive experiments demonstrate the superiority of our approach over state-of-the-art methods, achieving substantial improvements of **3.8%** and **1.7%** on average across all metrics on the challenging benchmark datasets, Charades and MultiTHUMOS. Our code will be released upon paper publication.*

1. Introduction

Dense action detection aims to recognize and temporally localize all actions within an untrimmed video, even when the actions occur concurrently. A deep understanding of these complex action semantics is crucial for many real-world applications, such as autonomous driving, sports analytics, and complex surveillance, where actions are rarely isolated.

(a) Current Approaches



(b) Our Approach (RefDense)

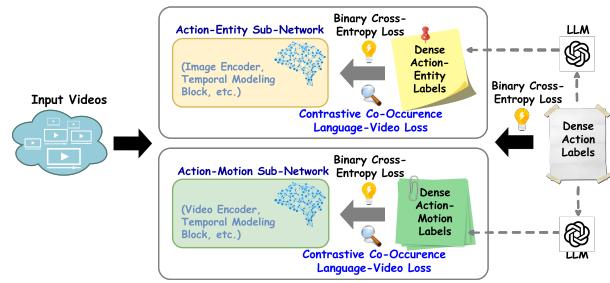


Figure 1. Comparison of current approaches and our proposed approach, RefDense, for tackling the dense action detection task. (a) Current approaches directly address the entire problem (i.e., detecting dense, ambiguous actions) using a single network, optimized solely with Binary Cross-Entropy (BCE) loss. In contrast, (b) RefDense decomposes the task into two sub-tasks (i.e., detecting dense, unambiguous entity and motion sub-concepts underlying the actions classes) and assigns them to distinct sub-networks. Furthermore, our approach is optimized using both BCE loss and our proposed contrastive co-occurrence language-video loss.

To tackle this task, current approaches [6, 8, 9, 21, 24, 28] typically follow a common pipeline. First, the video features are extracted using a pre-trained Encoder (e.g., I3D [3], CLIP [19]). Then, the features are fed into a Temporal Modeling Block (e.g., multi-scale transformer [6, 21], GNN [4, 8]) to capture temporal relationships among the segments, followed by a classification head that maps the learned representations to multi-action probabilities, enabling dense action detection. Finally, the entire network is optimized using binary cross-entropy (BCE) loss.

In dense action detection, beyond the challenge of temporal overlaps, action classes often exhibit semantic overlap

(*i.e.*, class ambiguity). This overlap can arise from shared entities or motion that define the action classes. For example, in the MultiTHUMOS dataset [27], the action classes “Hammer Throw Wind Up” and “Hammer Throw Spin” share an identical entity, a hammer. Similarly, in the Charades dataset [22], the classes “Holding a Bag” and “Holding a Sandwich” overlap in motion, the act of holding. We argue that the dual challenge of handling temporal and action class overlaps is too complex to be effectively addressed by the traditional dense action detection pipeline. This motivated us to raise a novel question: **Can we reduce the problem’s complexity by eliminating the class overlaps, thereby enabling the network to focus solely on resolving temporal overlaps?** To achieve this, we introduce a paradigm shift in solving this task. Instead of directly detecting dense, ambiguous actions using a single network, we propose decomposing the task into detecting dense, unambiguous sub-concepts underlying the action classes (*i.e.*, entity and motion sub-concepts), and assigning these sub-tasks to distinct sub-networks. By isolating the unambiguous components of actions, each sub-network can focus exclusively on resolving a single challenge, dense temporal overlaps.

To implement this novel paradigm, we (i) design a network comprising two sub-networks, Action-Entity and Action-Motion, and (ii) decompose dense action labels into dense action-entity and dense action-motion labels using prompts and a pre-trained large language model (LLM). While both sub-networks receive the same input video, Action-Entity focuses solely on detecting dense entity concepts involved in dense actions, whereas Action-Motion is dedicated to detecting dense motion concepts involved in dense actions. The dense temporal entity and motion representations learned by the sub-networks are then concatenated for dense action detection. The entire network is optimized using the original dense action labels and the BCE loss, while the Action-Entity and Action-Motion sub-networks are individually optimized using the dense action-entity and dense action-motion labels with the BCE loss.

In dense action detection, where multiple concepts can occur simultaneously, awareness of class dependencies can significantly enhance performance. For instance, in scenarios like cooking, actions such as “Pouring” and “Stirring” often occur together. However, we argue that the current dense action detection networks [6, 21, 24, 28] cannot effectively learn the relationships among the co-occurrence classes as they are trained using the BCE loss which treats each action class independently during the optimization process. This limitation motivates us to raise our second novel question: **Can we improve network optimization to fully unlock the potential benefits of co-occurring concepts?** To achieve this, we propose providing explicit supervision on co-occurring concepts in the input

video during network optimization. Inspired by contrastive language-image pretraining [19], we introduce Contrastive Co-occurrence Language-Video learning, which aligns the video features in the embedding space with the textual features of all co-occurring classes. Specifically, we assign a textual description to each co-occurring concepts in the input video and use a frozen pre-trained text encoder to extract their features. Then, we adapt the noise contrastive estimation loss to match the video features with the text features of all co-occurring classes. Through this, the network not only receives explicit knowledge of co-occurring concepts during training, but also implicitly benefits from the learned semantics of related concepts within the embedding space of pre-trained language models.

In Fig. 1, we compare current approaches to our proposed method (RefDense) in tackling the dense action detection task.

Our key contributions are summarized as follows: (i) we introduce a paradigm shift in solving dense action detection task—decomposing the problem complexity for the network—an approach that can also benefit solving other dense computer vision problems (*e.g.*, dense captioning); (ii) we pioneer the first exploration of explicitly addressing action class ambiguities in dense action detection task; (iii) for the first time, we introduce an optimization process for dense action detection, which enables the network to leverage explicit supervision on co-occurring concepts during training. This approach can enhance the optimization process of any existing or future dense action detection network; (iv) our comprehensive comparison using multiple metrics and challenging benchmark datasets against state-of-the-art approaches demonstrates the superiority of our method, *e.g.*, achieving substantial improvements of **3.8%** and **1.7%** on average across all metrics on Charades and MultiTHUMOS, respectively, and (v) our extensive ablation studies on these benchmark datasets, evaluated across multiple metrics, highlight the effectiveness of each component in our method’s design.

2. Related Works

Dense Action Detection – Current dense action detection approaches [8, 9, 21, 24, 28] typically follow a common pipeline. First, the video is divided into segments, and a frozen, pre-trained Encoder (*e.g.*, I3D, CLIP) extracts features from each segment. These features are then passed to a Temporal Modeling Block that captures their temporal relationships, followed by a classification layer that maps the learned representations to multi-action probabilities. The network is optimized using BCE loss. Although most of the pipeline is shared across approaches, the primary distinctions lie in the design of the Temporal Modeling Block. Below, we briefly review this block in existing approaches.

Pre-transformer approaches, such as [13, 17, 18], rely on

Gaussian or convolutional filters to represent a video as a sequence of multi-activity events. While these methods are effective at modeling short, dense actions, the inherent temporal limitations of Gaussian and convolutional kernels restrict their ability to capture longer actions. With the success of transformers in modeling long-term dependencies, several works [5, 6, 8, 21, 24, 28] have developed transformer-based networks. Among these, some approaches, such as [6, 21, 23, 28], focus on modeling various ranges of temporal relationships using multi-scale transformer networks or DETR-based architectures [2]. On the other hand, Tirupattur et al. [24] introduce the concept of benefiting from learning co-occurrence class relationships. To learn these relationships, they propose explicitly modeling all action classes within the network architecture. Similarly, Dai et al. [8] embed all objects in the dataset into the network’s architecture. However, not only do their designs lack computational efficiency due to their dependence on the maximum number of classes, but they also fail to fully capture co-occurrence relationships despite explicitly modeling the classes, as the networks are still optimized using the BCE loss, which treats each class independently. To the best of our knowledge, for the first time, our proposed contrastive co-occurrence language-video loss, is designed to overcome this limitation in network optimization by providing explicit supervision on co-occurring concepts during training. Furthermore, as it is a general loss function applied in the embedding space, it can benefit the optimization process in any existing or future network.

Although transformer-based approaches show performance improvements over traditional methods, the inherent complexity of handling both temporal and action class overlaps poses a substantial obstacle for networks. We address this by eliminating one of the overlaps; we propose to decompose the task of detecting dense ambiguous actions to detecting dense non-ambiguous sub-concepts underlying the action classes, and assign these sub-tasks to distinct sub-networks. By isolating these non-ambiguous components, each sub-network focuses exclusively on resolving a single challenge, dense temporal overlaps.

Vision-Language for Action Detection – Building on CLIP’s zero-shot capabilities [19], many works, such as [10, 14–16], adapt its language-image pre-training paradigm for zero-shot or few-shot action detection. Following this, some works, such as [1, 26], explore using language models for network pre-training. In contrast, [8, 12] integrate language models directly during training. For instance, Dai et al. [8] introduce an object-centric graph for indoor activity detection and leverage language supervision to ensure that each graph node corresponds to a distinct object, while [12] use language to obtain pseudo-labels for weakly supervised learning. In a similar spirit, we benefit from language models during training. However, our goal is different from that

of prior works; we aim to leverage language to effectively learn the relationships among co-occurring concepts.

3. Methodology

In this section, we first define the dense action detection task and briefly review the common pipeline used by current approaches to tackle this task, focusing specifically on the multi-scale transformer-based approach presented in [21], which serves as the backbone of part of our network. We then elaborate on our proposed approach, RefDense.

3.1. Preliminaries

Problem Definition – In the dense action detection task, the goal is to identify all actions occurring at each timestamp of an untrimmed video, as described in [6, 21, 24, 28]. Given an untrimmed video sequence $V = \{I_n \in \mathbb{R}^{W \times H \times 3}\}_{n=1}^N$ of length N , each timestamp n has a multi-action class label $Y_n = \{y_{n,c} \in \{0, 1\}\}_{c=1}^C$, where C represents the total number of action classes in the dataset, and the set of action labels for the entire video is denoted as $Y = \{Y_n\}_{n=1}^N$. The network’s task is to estimate multi-action class probabilities $P = \{P_n\}_{n=1}^N$, where $P_n = \{p_{n,c} \in [0, 1]\}_{c=1}^C$.

Current Pipeline to Tackle Dense Action Detection – To tackle dense action detection task, the most widely used pipeline consists of three main components: an Encoder, a Temporal Modeling Block, and a Classification layer. The Encoder, typically a frozen pre-trained 3D CNN, processes the input video sequence V for the Temporal Modeling Block. First, the video is divided into non-overlapping K -frame video segments $V = \{S_t\}_{t=1}^T$, where $S_t \in \mathbb{R}^{K \times W \times H \times 3}$ and $T = \frac{N}{K}$. The segments are then fed into the Encoder to obtain segment-level input tokens

$$F = \{\text{Encoder}(S_t)\}_{t=1}^T, \quad (1)$$

where $F \in \mathbb{R}^{T \times D}$. The Temporal Modeling Block receives the input video tokens to exploit a range of temporal relationships among them. For example, in the recent state-of-the-art method presented in [21], the Temporal Modeling Block is a multi-scale transformer, where a self-attention module is first applied to all input tokens to learn a fine-grained temporal representation of the video

$$\bar{F}_{fin} = \text{Self-attn}_\Theta(F), \quad (2)$$

where $\bar{F}_{fin} \in \mathbb{R}^{T \times D^*}$. Then, M convolutional layers with different strides are applied to the fine-grained features to down-sample them $\acute{F}_{crs,\theta} = \text{Conv}_\theta(\bar{F}_{fin})$, where $\acute{F}_{crs,\theta} \in \mathbb{R}^{\frac{T}{2^\theta} \times D^*}$ and $\theta \in \{1, 2, \dots, M\}$. Subsequently, M self-attention modules are applied to further exploit the temporal dependencies among the coarsely down-sampled features

$$\acute{F}_{crs,\theta} = \text{Self-attn}_\theta(\acute{F}_{crs,\theta}), \quad (3)$$

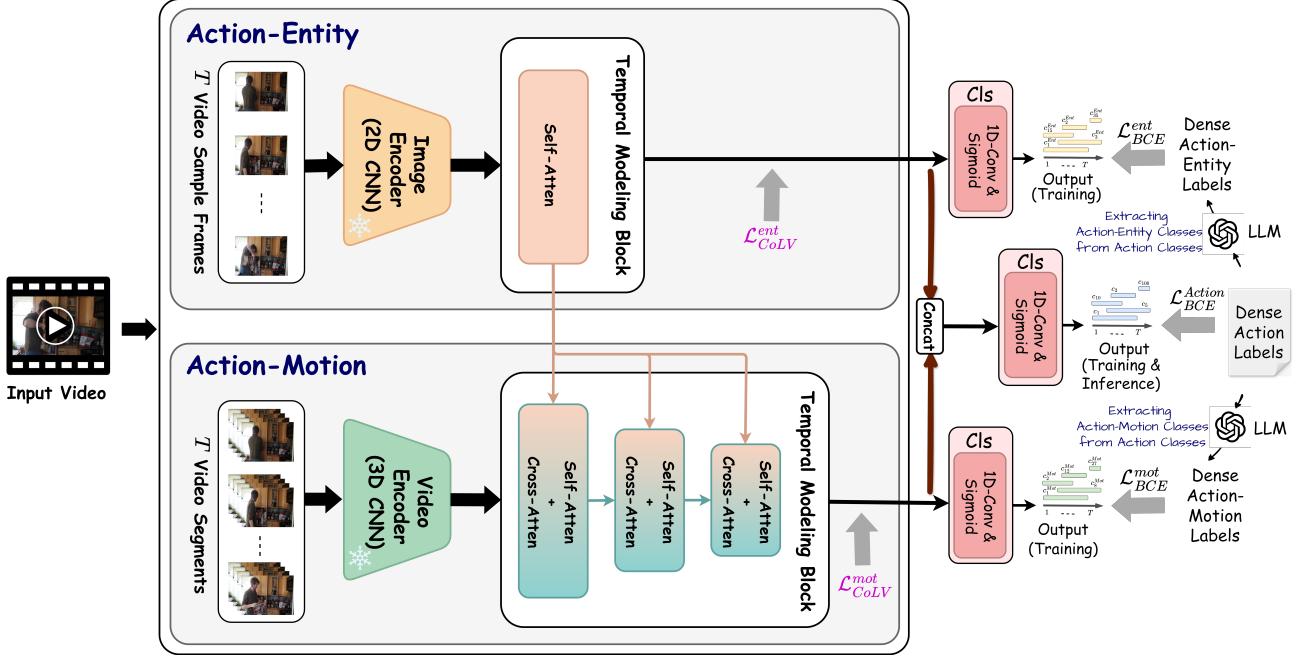


Figure 2. The overall scheme of RefDense. Our proposed network consists of two sub-networks: Action-Entity and Action-Motion. Action-Entity learns dense entity concepts associated with the action classes, while Action-Motion focuses on learning dense motion concepts related to the action classes. The entire network is optimized using the dense action labels and the BCE loss ($\mathcal{L}_{BCE}^{Action}$). Additionally, the sub-networks are optimized using dense action-entity and action-motion labels, which are derived from action labels, along with the BCE loss (\mathcal{L}_{BCE}^{ent} , and \mathcal{L}_{BCE}^{mot}) and our proposed contrastive co-occurrence language-video loss (\mathcal{L}_{CoLV}^{ent} and \mathcal{L}_{CoLV}^{mot}).

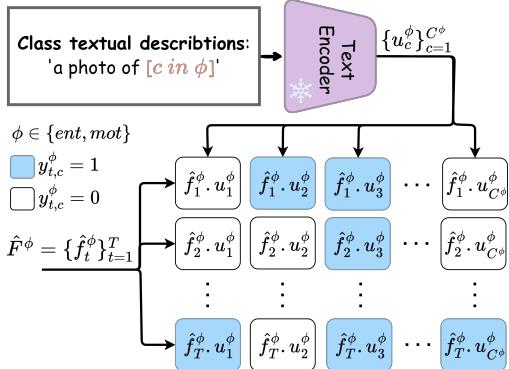


Figure 3. Alignment of temporal video features with textual features of co-occurring class concepts in our contrastive co-occurrence language-video loss (i.e., \mathcal{L}_{CoLV}^{ent} and \mathcal{L}_{CoLV}^{mot}).

where $\bar{F}_{crs,\theta} \in \mathbb{R}^{\frac{T}{2\theta} \times D^*}$. The coarse features are then up-sampled through linear interpolation $\bar{\bar{F}}_{crs,\theta} = \text{UpSample}(\bar{F}_{crs,\theta})$, where $\bar{\bar{F}}_{crs,\theta} \in \mathbb{R}^{T \times D^*}$ to match the same temporal length as the original input tokens. The up-sampled features, together with the fine-grained ones, are fused using techniques such as summation or concatenation for action classification $\hat{F} = \text{Fuse}(\bar{F}_{fin}, \bar{\bar{F}}_{crs,1}, \dots, \bar{\bar{F}}_{crs,M})$, where $\hat{F} \in \mathbb{R}^{T \times D^*}$.

Finally, the Classification layer, typically composed of fully connected or 1D convolutional filters, produces the multi-

action class probabilities $P = \text{Cls}(\hat{F})$, where $P \in \mathbb{R}^{T \times C}$. The entire network is typically optimized using the ground truth labels Y and the BCE loss, $\mathcal{L}_{BCE} = \text{BCE}(Y, P)$ as

$$\text{BCE}(Y, P) = -\frac{1}{T} \sum_{t=1}^T \sum_{c=1}^C \ell_{bce}(y_{t,c}, p_{t,c}), \quad (4)$$

where $\ell_{bce}(y, p) = y \log(p) + (1-y) \log(1-p)$.

3.2. Reframing Dense Action Detection (RefDense)

We introduce a paradigm shift in solving the dense action detection task. Instead of tackling the entire complex problem—handling the dual challenge of temporal and action class overlaps (i.e., class ambiguity)—with a single network, we propose decomposing the task into less complex sub-tasks: detecting dense, unambiguous sub-concepts underlying action classes (i.e., entity and motion sub-concepts) and assigning these sub-tasks to distinct sub-networks. By isolating these unambiguous concepts of actions, the sub-networks can focus exclusively on resolving a single challenge—dense temporal overlaps.

To implement our proposed paradigm, we (i) design a network comprising two sub-networks: Action-Entity and Action-Motion, and (ii) decompose dense action labels into dense action-entity and dense action-motion labels using prompts and a pre-trained Large Language Model (LLM), as shown in Fig. 2. While both sub-networks receive the

same input video, Action-Entity is tailored to detect dense entity concepts, whereas Action-Motion is designed to detect dense motion concepts. The dense temporal entity and motion representations learned by the sub-networks are then concatenated for dense action detection. The entire network is optimized using dense action labels and the BCE loss, while the Action-Entity and Action-Motion sub-networks are also individually optimized using dense action-entity and dense action-motion labels, respectively, with the BCE loss. Furthermore, to effectively leverage the interrelationships among co-occurring concepts within the video, we optimize the network’s embedding space using our proposed contrastive co-occurrence language-video loss. In the following, we detail our network, label decomposition, and loss functions.

Dense Action-Entity & Dense Action-Motion Labels – These labels are extracted for each input video from its original action labels. First, a set of action-entity and action-motion classes is defined from all action classes using specific prompts and a pre-trained LLM, GPT-4 (see supp. file for details). For example, from the action class “Weight Lifting Clean”, the action-entity class “Barbell” and the action-motion class “Lifting-Clean” are extracted, respectively. Then, for each input video, using its corresponding action ground-truth label Y and the newly defined classes, we generate its dense action-entity and dense motion-entity labels as $Y^{ent} \in \mathbb{R}^{T \times C^{ent}}$ and $Y^{mot} \in \mathbb{R}^{T \times C^{mot}}$, where C^{ent} and C^{mot} refer to the number of defined action-entity and action-motion classes, and $C^{ent}, C^{mot} \leq C$.

Note: (i) The decomposed labels preserve the temporal boundaries of the original labels extracted from, and (ii) not all actions involve both entities and motion components (e.g., “Walking”). For actions with only one component, the label is generated only for that component.

Action-Entity Sub-Network – This sub-network aims to detect dense entity concepts involved in the action classes. To achieve this, it consists of two components: an Image Encoder and a Temporal Modeling Block. First, the Image Encoder, a frozen pre-trained 2D CNN, is applied to the sampled frames of the input video $\{I_t\}_{t=1}^T$ to extract their spatial features $\mathbb{F} = \{\text{ImageEncoder}(I_t)\}_{t=1}^T$, where $I_t \in \mathbb{R}^{W \times H \times 3}$, and $\mathbb{F} \in \mathbb{R}^{T \times D}$. For sampling, the middle frame of each video segment S_t is selected. Then, the Temporal Modeling Block, which employs a lightweight transformer, including a few self-attention layers, receives the spatial features to model the dense action-entity concepts.

$$\hat{F}^{ent} = \text{Self-att}_\Delta(\mathbb{F}), \text{ where } \hat{F}^{ent} \in \mathbb{R}^{T \times D^*}. \quad (5)$$

Action-Motion Sub-Network – The goal of this sub-network is to detect dense motion concepts involved in the action class. To achieve this, it consists of two components: a Video Encoder and a Temporal Modeling Block. First, the

Video Encoder, a frozen pre-trained 3D CNN, is applied on the video segments to extract their spatio-temporal video features as in Eq. 1, $F = \{\text{VideoEncoder}(S_t)\}_{t=1}^T$. The features are then processed through the Temporal Modeling Block. Unlike the dense entity concepts, which can be modeled using only fine-grained temporal video features with a lightweight transformer, learning dense motion is more challenging, as a motion concept can vary in duration across different video samples. For example, the motion concept “Holding” may last only a few seconds in one video but may take up to a minute in another. To address this, it is necessary to model multiple scales of temporal information. Therefore, to implement the Temporal Modeling Block in this sub-network, we use the multi-scale transformer proposed in [21] as backbone. However, in contrast to the original network, which uses only the self-attention mechanism to learn fine and coarse video representations, our Temporal Modeling Block benefits additionally from the guidance of the Action-Entity sub-network through a cross-attention mechanism. Eq. 2 and Eq. 3 are adapted as

$$\bar{F}_{fin}^{mot} = \text{Cross-attn}_\Theta(\text{Self-att}_\Theta(F), \hat{F}^{ent}), \quad (6)$$

$$\bar{F}_{crs,\theta}^{mot} = \text{Cross-attn}_\theta(\text{Self-att}_\theta(\bar{F}_{crs,\theta}), \hat{F}^{ent}). \quad (7)$$

This guidance enables the Action-Motion sub-network to focus more effectively on regions informed by the learned entity concepts. In term $\text{Cross-attn}(a, b)$, the Query is generated from a , and the Key and Value are derived from b . Finally, similar to the backbone [21], the dense motion video representations \hat{F}^{mot} are obtained by fusing the fine and coarse motion representations.

Sub-Networks Fusion for Dense Action Detection – To perform dense action detection, the dense entity and motion video representations learned by the sub-networks are first concatenated to form the full video representation. Then, a 1D convolutional filter is applied to the full features to predict multi-action probabilities for all video segments:

$$P = \text{Sig}(1\text{D-Conv}_\vartheta([\hat{F}^{ent}; \hat{F}^{mot}])), \quad (8)$$

where $[;]$ denotes the concatenation operation, Sig refers to the sigmoid activation function, and $P \in \mathbb{R}^{T \times C}$.

Binary Cross-Entropy Optimization – With the action probabilities P and action labels Y , the entire network is optimized using $\mathcal{L}_{BCE}^{Action} = BCE(Y, P)$. The Action-Entity and Action-Motion sub-networks are also individually optimized using BCE and dense action-entity and action-motion labels Y^{ent} and Y^{mot} . To perform this, during training, a 1D convolutional layer is added on top of each sub-network to obtain multi-entity and multi-motion probabilities

$$P^\phi = \text{Sig}(1\text{D-Conv}_\phi(\hat{F}^\phi)), \quad (9)$$

where $P^\phi \in \mathbb{R}^{N \times C^\phi}$ and $\phi \in \{ent, mot\}$. With the probabilities and labels, the network is optimized using $\mathcal{L}_{BCE}^{RD} = \sum_\phi \mathcal{L}_{BCE}^\phi$, where $\mathcal{L}_{BCE}^\phi = BCE(Y^\phi, P^\phi)$.

Contrastive Co-Occurrence Language-Video Learning

– In scenarios where multiple concepts occur simultaneously, awareness of class dependencies can enhance the method’s performance. However, we argue that optimizing with the BCE loss does not allow networks to effectively learn these relationships, as BCE treats each class label independently. To address this limitation, we propose providing explicit supervision on co-occurring concepts in the video during training. To achieve this, inspired by contrastive language-image pre-training [19], we align the learned video representations in the embedding space $\hat{F}^\phi = \{\hat{f}_t^\phi\}_{t=1}^T$ with the extracted text features of all co-occurring classes in the input video (see Fig. 3). Specifically, a textual sentence is assigned to each class occurring in the video as $txt_c^\phi = ‘a photo of [c in \phi]’$, where $[c in \phi]$ represents the text description for class c within the class set ϕ , and $\phi \in \{ent, mot\}$. Then, a frozen pre-trained Text Encoder (e.g., CLIP’s text encoder) is then used to extract their features $u_c^\phi = \text{TextEncoder}(txt_c^\phi)$. Finally, the noise contrastive estimation is adapted to match the visual representations of each video segment with the text features of all the co-occurring concepts in that segment as:

$$\mathcal{L}_{CoLV}^{RD} = \sum_\phi \mathcal{L}_{CoLV}^\phi, \quad (10)$$

$$\mathcal{L}_{CoLV}^\phi = -\frac{1}{T} * \sum_{t=1}^T \frac{1}{|\beta(t)^\phi|} \sum_{e \in \beta(t)^\phi} \log \frac{\exp(\hat{f}_t^\phi \cdot u_e^\phi / \tau)}{\sum_{\substack{c=1, \\ c \notin \beta(t)^\phi}}^C \exp(\hat{f}_t^\phi \cdot u_c^\phi / \tau)}, \quad (11)$$

$$\beta(t)^\phi = \{e \mid e \in \{1, 2, \dots, C^\phi\}, y_{t,e}^\phi = 1\}. \quad (12)$$

Through this, the network not only receives explicit knowledge of co-occurring concepts, but also implicitly benefits from the learned semantic among related concepts within the embedding space of pre-trained language models.

4. Experimental Results

Datasets – We evaluate our proposed approach on the primary benchmark datasets for this task, Charades [22] and MultiTHUMOS [27], as well as on the recent TSU dataset [7], with results and details included in the supp. file. Charades is a large-scale dataset containing 9,848 videos of daily activities across 157 action classes, with a high degree of temporal overlap among action instances. MultiTHUMOS, the dense multi-label version of the single-label action detection dataset THUMOS’14 [11], includes 413 long sports activity videos across 65 action classes. Charades and MultiTHUMOS considered very challenging datasets as with the current state-of-the-art mean Average Precision (mAP) reaching only 32.0% and 45.5%.

Implementation Details – For the Video Encoder, following previous works [13, 21, 24], we use the pre-trained I3D network [3]. For the Image Encoder and Text Encoder, we employ CLIP’s pre-trained ResNet-50 image encoder and CLIP’s text encoder [19]. In Action-Entity, we implement the Temporal Modeling Block using one position-aware self-attention block as designed in [21]. In Action-Motion, we utilize the multi-scale transformer PAT [21] as the backbone for the Temporal Modeling Block. For more details on the network architecture, see the Supp. file. The length of each video segment is set to $K = 8$ frames. During training, $T = 256$ consecutive video segments are randomly sampled from an untrimmed video sequence to serve as network input. At inference, we follow previous work [13, 21, 24] and make predictions on the full video sequence. For Charades, 38 action-entity and 38 action-motion classes are defined, while for MultiTHUMOS, 28 action-entity and 50 action-motion classes are defined.

We conducted our experiments using PyTorch on an NVIDIA GeForce RTX 3090 GPU. Our model was trained with the Adam optimizer, starting with an initial learning rate of 0.0001. We used a batch size of 5 for 25 epochs and a batch size of 3 for 300 epochs and for Charades and MultiTHUMOS, respectively. The learning rate was reduced by a factor of 10 after every 7 epochs for Charades and after every 130 epochs for MultiTHUMOS. Note that the different training settings for Charades and MultiTHUMOS are due to their varying sizes.

4.1. State-of-the-Art Comparison

In this section, we compare the performance of our approach with current state-of-the-art methods using different metrics. Note: Here, our results and comparisons are based on RGB input features. However, results and comparisons incorporating RGB and optical flows can be found in the supp. file.

The primary metric for dense action detection task is the standard per-frame mAP. Table 1 presents comparative results on Charades and MultiTHUMOS using this metric. The results demonstrate the superiority of our approach over state-of-the-art methods, achieving significant improvements of **1.4%** and **1.1%** mAP on Charades and MultiTHUMOS, respectively, which corresponds to a substantial relative improvement of **4.4%** and **2.5%** over the current best-performing methods. Furthermore, the results reveal that our approach exhibits better generalization across different datasets, with the smallest performance difference between the two datasets (13.2%). In contrast, other methods are less consistent; for example, DualDET [28] shows a gap of 22.3%, and ANN [8] is limited to indoor activity datasets due to its object-centric architecture.

The standard mAP assesses the performance by evaluating each class independently. However, it does not

Method	GFLOPs	mAP(%)	
		Charades	MultiTHUMOS
R-C3D [25]	ICCV 2017	-	12.7
SuperEvent [18]	CVPR 2018	0.8	18.6
TGM [17]	ICML 2019	1.2	20.6
PDAN [5]	WACV 2021	3.2	23.7
CoarseFine [13]	CVPR 2021	-	25.1
MLAD [24]	CVPR 2021	44.8	18.4
CTRN [4]	BMVC 2021	-	25.3
PointTAD [23]	NeurIPS 2022	-	21.0
MS-TCT [6]	CVPR 2022	6.6	25.4
PAT [21]	ICCVW 2023	8.5	26.5
TTM [20]	CVPR 2023	-	28.8
ANN [8]	BMVC 2023	-	32.0
DualDET [28]	CVPR 2024	5.5	23.2
RefDense	11.5	33.4 (+1.4)	46.6 (+1.1)

Table 1. Dense action detection results on the Charades and MultiTHUMOS datasets using RGB inputs, in terms of per-frame mAP. The best and the second best results are in **Bold** and underlined.

explicitly measure whether models learn the relationships amongst the classes. To overcome this, [24] introduce a set of action-conditional metrics, including action-conditional mean Average Precision mAP_{ac} , action-conditional F1-Score $F1_{ac}$, action-conditional Precision P_{ac} , and action-conditional Recall R_{ac} . These metrics aim to explicitly assess how well pairwise class/action dependencies are modeled, both within a single frame and across different frames. Table 2 presents the comparative results on Charades and MultiTHUMOS using action-conditional metrics. While these metrics evaluate a method’s performance more effectively than standard mAP, only a few methods report results using them, primarily on Charades. Therefore, for a comprehensive comparison, we produced the results of previous methods under these metrics, using RGB inputs, with their publicly available code whenever accessible. We hope this comprehensive comparison on two datasets benefits the community in future works. Table 2 demonstrates the superiority of our method over current state-of-the-art approaches in detecting dense actions. Specifically, it achieves an average improvement of **4.1%** on Charades and **1.8%** on MultiTHUMOS across all conditional metrics.

Qualitative Comparison – In Fig. 4, we qualitatively compare our approach with the state-of-the-art methods PAT [21] and MS-TCT [6] on a test video sample of Charades. The results show that not only do the action predictions of our method have better overlap with the ground truth labels than the other methods, but it also detects more action classes, *i.e.*, Our method detects 4 action types, while PAT and MS-TCT detect 3 and 2 action types, respectively.

4.2. Ablation Studies

In this section, we extensively evaluate the impact of key components of our proposed approach using both type of metrics and on two datasets. Note, to perform these exper-

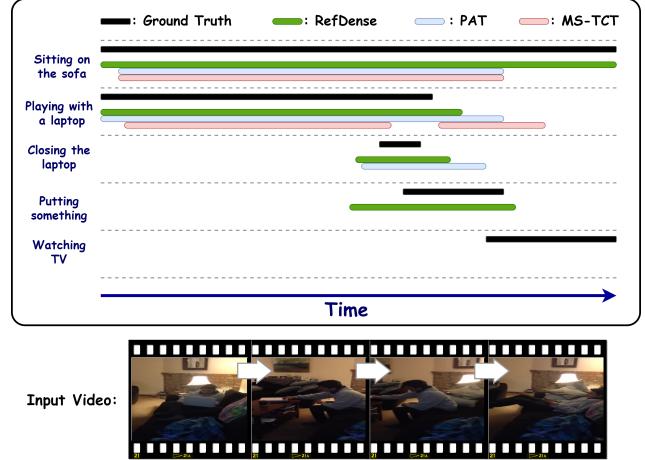


Figure 4. Qualitative comparison with previous approaches (PAT [21] and MS-TCT [6]) on a test video sample of Charades.

iments all action conditional metrics are measured over a temporal window of size $\tau = 0$.

How Sub-Networks Tackle the Task Independently – Table 3 compares the performance of our approach with its individual sub-networks, each challenged to independently solve the entire complex problem (*i.e.*, handing both temporal and class overlaps). To ensure a fair comparison, we adapted these baselines to utilize I3D and CLIP image features through feature concatenation. The findings in Table 3 demonstrates that our approach significantly surpasses its sub-networks across both datasets and all metrics. Notably, it surpasses Action-Entity by an average of **5.6%** on Charades and **2.9%** on MultiTHUMOS, and Action-Motion by **4.7%** on Charades and **1.3%** on MultiTHUMOS.

Impact of Sub-Labels – The goal of decomposed labels, dense action-entity and action-motion labels, is to ensure that each sub-network focuses solely on its own sub-task. The results in Table 4 confirm their essential role in addressing the problem. Removing these labels from training leads to a significant performance drop, with an average decrease of **2.6%** on Charades and **1.6%** on MultiTHUMOS.

Impact of \mathcal{L}_{CoLV}^{RD} – We ablate our proposed contrastive co-occurrence language-video loss \mathcal{L}_{CoLV}^{RD} in Table 5. The results indicate that providing explicit supervision on co-occurring concepts through our loss significantly enhances the method’s performance, over **1.0%** improvement, across all metrics. Notably, this improvement is achieved purely through optimization, without modifying the network.

Generalization of \mathcal{L}_{CoLV}^{RD} – Our proposed loss, \mathcal{L}_{CoLV}^{RD} , is a general loss function that can be applied to the embedding space of any existing or future network to improve their optimization. For, example, in Table 6, we present its impact when applied to the existing PAT network [21]. The results show that it enhances PAT’s performance across all metrics. Specifically, the standard mAP and mAP_{ac} increase signif-

Method	Charades								MultiTHUMOS							
	$\tau = 0$				$\tau = 20$				$\tau = 0$				$\tau = 20$			
	mAP _{ac}	F1 _{ac}	P _{ac}	R _{ac}	mAP _{ac}	F1 _{ac}	P _{ac}	R _{ac}	mAP _{ac}	F1 _{ac}	P _{ac}	R _{ac}	mAP _{ac}	F1 _{ac}	P _{ac}	R _{ac}
MLAD [24]	28.4	12.5	21.7	8.6	34.7	13.6	21.0	10.1	18.0	29.4	28.8	13.0	19.6	30.5	31.4	14.2
CTRNN [4]	29.7	11.9	23.9	8.1	36.8	12.9	27.1	9.1	-	-	-	-	-	-	-	-
MS-TCT [6]	29.4	14	24.8	9.7	35.1	15.4	24.3	11.1	26.3	33.5	33.8	21.5	28.8	35.4	37.8	22.8
PAT [21]	30.0	27.1	25.9	28.4	36.3	30.2	28.9	31.7	29.1	35.0	33.5	25.7	31.4	37.5	37.3	27.1
ANN [8]	35.4	20.4	31.4	-	41.8	22.3	30.4	-	-	-	-	-	-	-	-	-
RefDense	37.5	33.0	32.0	33.9	43.7	36.4	35.4	37.4	31.1	37.1	35.5	27.7	33.1	39.2	38.8	28.8
	(+2.0)	(+5.9)	(+0.6)	(+5.5)	(+1.9)	(+6.2)	(+5.0)	(+5.7)	(+2.0)	(+2.1)	(+1.7)	(+2.0)	(+1.7)	(+1.7)	(+1.0)	(+1.7)

Table 2. Dense action detection results on Charades and MultiTHUMOS using RGB inputs, evaluated based on the action-conditional metrics with cross-action dependencies over a temporal window of size τ . The best and the second best results are in **Bold** and underlined.

Network	Charades			MultiTHUMOS		
	mAP	mAP _{ac}	F1 _{ac}	mAP	mAP _{ac}	F1 _{ac}
Action-Entity	27.7 (-5.7)	31.2 (-6.3)	28.4 (-4.6)	40.0 (-6.6)	28.5 (-2.6)	32.3 (-4.8)
Action-Motion	30.4 (-3.0)	35.0 (-2.5)	30.9 (-3.1)	44.4 (-2.2)	30.6 (-0.5)	36.0 (-1.1)
RefDense	33.4 (+2.5)	37.5 (+3.0)	33.0 (+2.1)	46.6 (+2.1)	31.1 (+1.3)	37.1 (+1.1)

Table 3. Ablation studies on network design.

Labels	Charades			MultiTHUMOS		
	mAP	mAP _{ac}	F1 _{ac}	mAP	mAP _{ac}	F1 _{ac}
Entity Motion						
X ✓	31.1	34.4	30.8	44.8	30.8	36.4
✓ X	31.5	35.4	31.3	44.6	29.8	36.2
X X	30.9	34.5	30.9	44.5	29.8	36.0
✓ ✓	33.4 (+2.5)	37.5 (+3.0)	33.0 (+2.1)	46.6 (+2.1)	31.1 (+1.3)	37.1 (+1.1)

Table 4. Ablation studies on employing sub-labels for training.

icantly, by **1.1%** on Charades and MultiTHUMOS, respectively. This improvement is achieved in an end-to-end manner without altering the network architecture.

Impact of Cross-Attention Mechanism – To enhance the learning of motion concepts, we design the Action-Motion sub-network to receive guidance from the Action-Entity sub-network through the cross-attention mechanism, allowing it to focus more on regions highlighted by the learned action-entity concepts. The results in Table 7 verify the effectiveness of this design, showing that by adding the cross-attention mechanism, we achieve over **1.0%** improvement across most metrics on both datasets.

5. Conclusion

In this paper, we introduce a paradigm shift in solving the dense action detection task. Instead of tackling the entire complex problem—handling the dual challenge of temporal and action class overlaps (*i.e.*, class ambiguity)—using a single network, we propose decomposing the task of detecting dense, ambiguous actions into detecting

\mathcal{L}_{CoLV}^{RD}	Charades			MultiTHUMOS		
	mAP	mAP _{ac}	F1 _{ac}	mAP	mAP _{ac}	F1 _{ac}
X	32.2	36.2	32.0	44.9	30.1	35.9
✓	33.4 (+1.2)	37.5 (+1.3)	33.0 (+1.0)	46.6 (+1.7)	31.1 (+1.0)	37.1 (+1.2)

Table 5. Ablation studies on \mathcal{L}_{CoLV}^{RD} .

	Charades			MultiTHUMOS		
	mAP	mAP _{ac}	F1 _{ac}	mAP	mAP _{ac}	F1 _{ac}
PAT [21]	25.6	30.0	27.1	44.6	29.1	35.0
PAT [21] + \mathcal{L}_{CoLV}^{RD}	26.7 (+1.1)	30.7 (+0.7)	27.7 (+0.6)	45.2 (+0.6)	30.2 (+1.1)	35.4 (+0.4)

Table 6. Impact of \mathcal{L}_{CoLV}^{RD} on PAT [21].

Cross-Attn	Charades			MultiTHUMOS		
	mAP	mAP _{ac}	F1 _{ac}	mAP	mAP _{ac}	F1 _{ac}
X	31.8	35.8	31.4	44.7	29.8	36.4
✓	33.4 (+1.6)	37.5 (+1.7)	33.0 (+1.6)	46.6 (+1.9)	31.1 (+1.3)	37.1 (+0.7)

Table 7. Ablation studies on using the cross-attention mechanism.

dense, unambiguous sub-concepts that define the action classes, and assigning these sub-tasks to distinct sub-networks. By isolating these unambiguous concepts, each sub-network can focus exclusively on resolving a single challenge—dense temporal overlaps. Furthermore, to effectively learn the relationships among co-occurring concepts in a video, we propose a novel contrastive language-guided loss that provides explicit supervision on co-occurring concepts during training. Our extensive experiments, conducted on the challenging benchmark datasets Charades and MultiTHUMOS using multiple metrics, demonstrate that our method significantly outperforms state-of-the-art approaches across all metrics. Additionally, ablation studies highlight the effectiveness of the key components of our method. Future work will extend our approach to dense multi-modal (*e.g.*, audio-visual) dense action detection.

References

- [1] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. LOCVTP: Video-Text Pre-training for Temporal Localization. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022. 3
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end Object Detection with Transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3
- [3] Joao Carreira and Andrew Zisserman. Quo vadis, Action Recognition? a New Model and the Kinetics Dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 6
- [4] Rui Dai, Srijan Das, and Francois Bremond. CTRN: Class-Temporal Relational Network for Action Detection. *British Machine Vision Conference*, 2021. 1, 7, 8
- [5] Rui Dai, Srijan Das, Luca Minciullo, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. PDAN: Pyramid Dilated Attention Network for Action Detection. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 2970–2979, 2021. 3, 7
- [6] Rui Dai, Srijan Das, Kumara Kahatapitiya, Michael S Ryoo, and Francois Bremond. MS-TCT: Multi-Scale Temporal ConTransformer for Action Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 20041–20051, 2022. 1, 2, 3, 7, 8
- [7] Rui Dai, Srijan Das, Saurav Sharma, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota Smarthome Untrimmed: Real-World Untrimmed Videos for Activity Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 2533–2550, 2022. 6
- [8] Rui Dai, Srijan Das, Michael Ryoo, and Francois Bremond. AAN: Attributes-Aware Network for Temporal Action Detection. In *British Machine Vision Conference*, 2023. 1, 2, 3, 6, 7, 8
- [9] Xiyang Dai, Bharat Singh, Joe Yue-Hei Ng, and Larry Davis. TAN: Temporal Aggregation Network for Dense Multi-Label Action Recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision*, pages 151–160. IEEE, 2019. 1, 2
- [10] Edward Fish, Jon Weinbren, and Andrew Gilbert. PLOT-TAL-Prompt Learning with Optimal Transport for Few-Shot Temporal Action Localization. *arXiv preprint arXiv:2403.18915*, 2024. 3
- [11] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. THUMOS Challenge: Action Recognition with a Large Number of Classes, 2014. 6
- [12] Chen Ju, Kunhao Zheng, Jinxiang Liu, Peisen Zhao, Ya Zhang, Jianlong Chang, Qi Tian, and Yanfeng Wang. Distilling Vision-Language Pre-training to Collaborate with Weakly-Supervised Temporal Action Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14751–14762, 2023. 3
- [13] Kumara Kahatapitiya and Michael S Ryoo. Coarse-Fine Networks for Temporal Activity Detection in Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8385–8394, 2021. 2, 6, 7
- [14] Zhiheng Li, Yujie Zhong, Ran Song, Tianjiao Li, Lin Ma, and Wei Zhang. DeTAL: Open-Vocabulary Temporal Action Localization with Decoupled Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3
- [15] Benedetta Liberatori, Alessandro Conti, Paolo Rota, Yiming Wang, and Elisa Ricci. Test-Time Zero-Shot Temporal Action Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18720–18729, 2024.
- [16] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-Shot Temporal Action Detection via Vision-Language Prompting. In *European Conference on Computer Vision*, pages 681–697. Springer, 2022. 3
- [17] AJ Piergiovanni and Michael Ryoo. Temporal Gaussian Mixture Layer for Videos. In *International Conference on Machine learning*, pages 5152–5161. PMLR, 2019. 2, 7
- [18] AJ Piergiovanni and Michael S Ryoo. Learning Latent Super-Events to Detect Multiple Activities in Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, 2018. 2, 7
- [19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 6
- [20] Michael S Ryoo, Keerthana Gopalakrishnan, Kumara Kahatapitiya, Ted Xiao, Kanishka Rao, Austin Stone, Yao Lu, Julian Ibarz, and Anurag Arnab. Token Turing Machines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19070–19081, 2023. 7
- [21] Faegheh Sardari, Armin Mustafa, Philip JB Jackson, and Adrian Hilton. PAT: Position-Aware Transformer for Dense Multi-Label Action Detection. In *Proceedings of the IEEE International Conference on Computer Vision - Workshop*, pages 2988–2997, 2023. 1, 2, 3, 5, 6, 7, 8
- [22] Gunnar A Sigurdsson, Gülcin Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *Proceedings of the European Conference on Computer Vision*, pages 510–526, 2016. 2, 6
- [23] Jing Tan, Xiaotong Zhao, Xintian Shi, Bin Kang, and Limin Wang. PointTAD: Multi-Label Temporal Action Detection with Learnable Query Points. In *Advances in Neural Information Processing Systems*, 2022. 3, 7
- [24] Praveen Tirupattur, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Modeling Multi-Label Action Dependencies for Temporal Action Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1460–1470, 2021. 1, 2, 3, 6, 7, 8

- [25] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5783–5792, 2017. [7](#)
- [26] Mengmeng Xu, Erhan Gundogdu, Maksim Lapin, Bernard Ghanem, Michael Donoser, and Loris Bazzani. Contrastive Language-Action Pre-training for Temporal Localization. *arXiv preprint arXiv:2204.12293*, 2022. [3](#)
- [27] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos. *International Journal of Computer Vision*, 126(2):375–389, 2018. [2](#), [6](#)
- [28] Yuhan Zhu, Guozhen Zhang, Jing Tan, Gangshan Wu, and Limin Wang. Dual DETRs for Multi-Label Temporal Action Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 18559–18569, 2024. [1](#), [2](#), [3](#), [6](#), [7](#)