JOADAA: joint online action detection and action anticipation

Mohammed Guermal Abid Ali Rui Dai[1] Francois Bremond

Inria, France[1]
Université Cote d'Azur, France
name.surname@inria.fr

Abstract

Action anticipation involves forecasting future actions by connecting past events to future ones. However, this reasoning ignores the real-life hierarchy of events which is considered to be composed of three main parts: past, present, and future. We argue that considering these three main parts and their dependencies could improve performance. On the other hand, online action detection is the task of predicting actions in a streaming manner. In this case, one has access only to the past and present information. Therefore, in online action detection (OAD) the existing approaches miss semantics or future information which limits their performance. To sum up, for both of these tasks, the complete set of knowledge (past-present-future) is missing, which makes it challenging to infer action dependencies, therefore having low performances. To address this limitation, we propose to fuse both tasks into a single uniform architecture. By combining action anticipation and online action detection, our approach can cover the missing dependencies of future information in online action detection. This method referred to as JOADAA, presents a uniform model that jointly performs action anticipation and online action detection. We validate our proposed model on three challenging datasets: THUMOS'14, which is a sparsely annotated dataset with one action per time step, CHARADES, and Multi-THUMOS, two densely annotated datasets with more complex scenarios. JOADAA achieves SOTA results on these benchmarks for both tasks.

1. Introduction

Envisioning upcoming occurrences plays a vital role in human intelligence as it aids in making choices while engaging with the surroundings. Humans possess an inherent skill to predict future happenings in diverse situations involving interactions with the environment. Likewise, the capacity to anticipate events is imperative for advanced AI systems operating in intricate settings, including interactions with other agents or individuals. The goal of online action detection (OAD) is to accurately pinpoint ongoing actions in streaming media, by predicting impending events. While action anticipation advances OAD and imitates the capacity of human cognition to anticipate events before they occur. Therefore, OAD and action anticipation are two important areas of research in computer vision, which have numerous applications in security surveillance, home-care, sports analysis, self-driving cars, and online danger detection. Human perception of actions can be viewed as a continuous cycle in which prior knowledge is used to forecast future behavior, and then present knowledge is used to revise and update future predictions. To tackle action detection, we propose a unified framework of action anticipation and online action detection. Our predictions are in two steps, first we anticipate up-coming actions based on past information. Second, we update the anticipation by introducing the present information. By doing so, we gain in the online action detection by introducing the anticipated actions as pseudo-future information. In addition, it improves the action anticipation by comparing the prediction to the present information, thus combining them to improve both tasks.

Transformer networks such as [1, 19, 26] have had a significant impact on computer vision and video understanding. This is due to their ability to capture long-range dependencies. LSTR [31], TesTra [35], or FUTR [13] have benefited from the transformer backbones to address the tasks of OAD and AA. However, OAD and AA (action anticipation) tasks suffer from limited information as they don't have access to future information and global knowledge of the scene. This limited information restricts the ability of transformers to capture long-range dependencies and to learn significant relations between events. This can be demonstrated by comparing the effectiveness of models for offline action detection with online action detection. Offline, one has access to all pieces of information and a clear knowl-

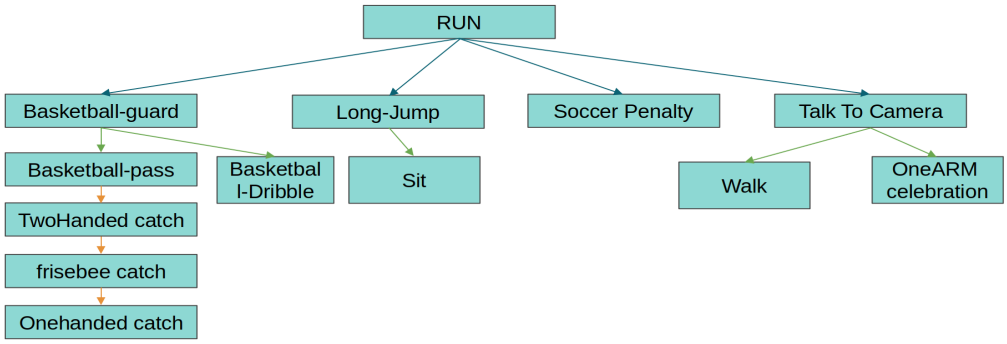

Figure 1. An example of human non-sequential dependencies. For instance, the actions *RUN and OneHanded Catch* are highly correlated but distant. Also the same start action ***RUN*** can lead to many different actions and scenarios. Therefore, it is very hard for online action detection or action anticipation to detect such relations without access to the future. In JOADAA, we propose to tackle this limitation by introducing a pseudo-future information by combining action anticipation and online action detection in the same task.

edge of the past, present, and future. Furthermore, complex densely annotated datasets (such as Multi-THUMOS [32]) have not been explored for online action detection and anticipation. It is challenging to recognize and foresee activities in such datasets. Most OAD architectures are only validated on sparsely-annotated activity datasets. Such simple annotated datasets are less challenging. First, these datasets do not have co-occurring actions. Second, they rarely have dependencies between actions in distant time steps. Furthermore, actions in densely annotated datasets have many possible outcomes. An example of these complex dependencies is given in Figure 1. Due to these challenges, OAD methods are only validated on simple datasets. Therefore, even with the help of transformers, it is difficult to build knowledge of these long-range dependencies without having access to complete information.

In the past, OAD and action anticipation have been treated as separate tasks. However, to tackle the above challenges, we propose JOADAA (Joint Online Action Detection and Action Anticipation) to tackle OAD and AA together. We create a pseudo-future when performing online action detection. By leveraging cross-attention between the real frame features and the anticipated frames, we enhance the quality of the features, thus improving the accuracy of the predictions by making the present aware of a pseudo-future. Next, we propose to extract two types of information from these updated features: Local dependencies using TCNs (temporal convolution networks) and global dependencies using MHA (multi-head attention). Finally, we fuse both pieces of information to make online action detection predictions.

In this paper, following previous work, we extract features from video clips using 3D convolution neural networks (3D CNNs). We use I3D [3] as a pre-trained backbone on the Kinetics dataset [16]. We store these extracted features in a memory bank. JOADAA consists of three main parts i) **Past Processing Block**, ii) **Anticipation prediction Block**, and iii) **Online action prediction Block**. First, we capture past information using a transformer encoder. The encoder output is first passed through a classification layer, which helps improve the quality of the embedding by making it class-dependent. Next, in the anticipation prediction part, we assume that we have not yet got the current frame. A transformer decoder is employed to learn from the last layer of the past embeddings to anticipate the upcoming actions in the next frame. This is carried out by introducing a set of learnable queries, called *anticipation queries*. Finally, the online action prediction part uses anticipation embedding and current frame features to enhance the quality of the current frame. The new enhanced present frame features are fused with past features. Finally, global and local information is extracted using MHA and TCN layers, respectively, achieving a new enhanced feature map. Based on the challenges discussed, we propose the following main contributions:

- We propose a new architecture **JOADDA**, to jointly perform online action detection and action anticipation.
- We tackle both tasks for two different types of datasets, a densely annotated dataset and a simple activity dataset.
- We validate our proposed method on three benchmark datasets and achieve new SOTA results for online action detection and action anticipation.

2. Related work

Online Action Detection is the task of localizing action instances in time steps. We distinguish two types of action detection i.e., offline and online. In off-line action detection, the model has access to the entire video [7, 22, 24, 29, 36]. Online action detection, on the other hand, occurs in real-time and has access to the past and the present only. RED [9] uses reinforcement loss to encourage early recognition of activities. IDN [8] learns discriminative features and stores only knowledge that is relevant in the present. To achieve optimal features, LAP-Net [21] presents an adaptive sampling technique. PKD [34] uses curriculum learning to transfer information from offline to online models. Shou et al. [23], similar to early action detection, focus on online detection of action start (ODAS). StartNet [10] divides ODAS into two stages and learns using a policy gradient. WOAD [11] employs video-level labeling and weakly-supervised learning. LSTR [31] uses a set of encoder-decoder architectures to capture the relations between long-term and short-term actions. They achieve state-of-the-art results on sparsely-annotated datasets but perform poorly on densely labeled datasets such as Multi-Thumas [32].

Action Anticipation is the task of predicting future actions given the limited observation of a video. In the past, many strategies have been proposed to solve the next action anticipation, forecasting a single future action in a matter of seconds. Recently, the idea of anticipating long-term activities from a long-range video has been put out. Girdhar and Grauman [12] introduced the anticipative video transformer (AVT), which anticipates the following action using a self-attention decoder, which was further improved by FUTR [13] for minutes-long future actions. However, their architecture is suitable only for simple activities and simple datasets, which is not applicable to real-world scenarios that have multiple actions occurring at the same time.

Finally, in the study of mixing action anticipation and online action prediction, the authors in [35] use the same architecture for both action anticipation and online action detection tasks. However, they dissociate these tasks, while we tackle both tasks jointly to improve both of them. Furthermore, the architecture in [35] is very similar to [31], therefore, the same limitations apply here as well.

In summary, to have adequate predictions, we need to build a well-descriptive hierarchy of information consisting of past, present, and future. Unfortunately, tasks such as online action detection or action anticipation do not have access to this global knowledge. In our work, we suggest combining OAD and AA in order to create pseudo-full knowledge that can improve action anticipation accuracy and produce comparable results for online action detection.

3. Proposed method

The whole architecture consists of three main parts, i) Past Processing Block, ii) Anticipation prediction Block, and iii) Online Action Prediction, as shown in Figure 2. First, a short-term past transformer-encoder enhances features. Second, an anticipation transformer-decoder anticipates the upcoming actions in the upcoming frames, using embedding output from the previous block and a set of learnable queries, which we call anticipation queries. Finally, a transformer-decoder uses the anticipation results and past information to predict the actions for the current frame (online action detection). Each module is explained in the following.

3.1. Past Processing Block

To enhance the ongoing action prediction, the initial stage in our model is to infer prior information. We employ a transformer encoder that accepts the embedding of previous frames as input. This enables us to highlight salient and robust frames by leveraging attention mechanisms, making our features more descriptive of previous activities (features). It can be challenging to identify which activity a person is performing solely based on the raw embedding or the current frame. For instance, if the current frame shows the person *holding a bottle*, we are not sure if the ongoing action will be *picking up the bottle, placing the bottle, drinking water, or pouring water*. However, if we know from the past that one of the previous actions was *opening the bottle*, we can be more confident that the person is more likely to *drink water*. These features are later used to anticipate future actions. Following [26], the equations below sum up the first block of our architecture:

$$F' = ATTENTION(F) \tag{1}$$

$$ATTENTION(F) = Softmax(QK^T/\sqrt{d_k})V \tag{2}$$

$$Q = W_q \times X, K = W_k \times X, V = W_v \times X \tag{3}$$

$$X = F + PE(F) \tag{4}$$

PE stands for positional encoding, and $F \in \mathbb{R}^{T \times D}$ are the extracted features using the pre-trained I3D model [3], and W_q, W_k and W_v are learnable weights.

Furthermore, we propose different approaches for the use of past information. Following [31] we use long-term and short-term past information. Experimentally, the use of long-term and short-term past information is highly dependent on the type of dataset. The first intuition is that more information is always good for a neural network as it provides a more detailed description of events in a video. Especially with the use of transformers, we can capture long-range dependencies to learn all the steps that lead to the current actions. However, in our study, we find that this is not always true. For instance, the very long-past knowledge

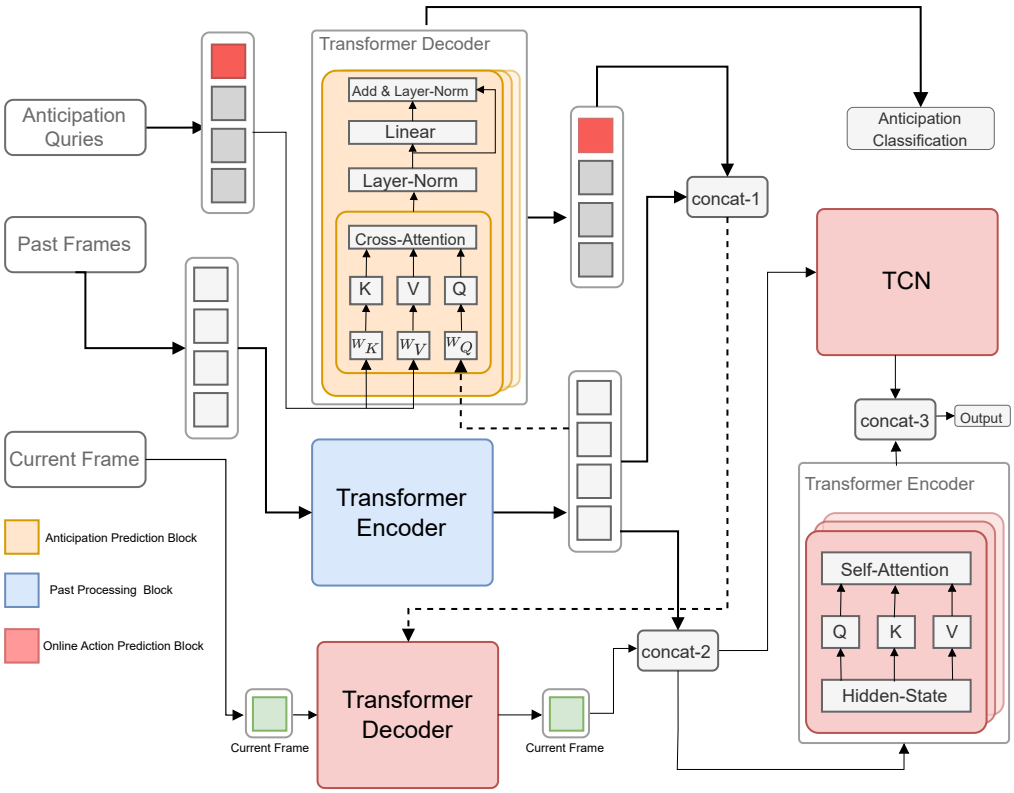

Figure 2. Proposed JOADAA architecture with three units i) Past processing, ii) Anticipation prediction, and iii) Online Action prediction. Each stage is highlighted by a color for better understanding. Each block will be explained in details in section 3

may sometimes harm performances, especially for densely annotated datasets. In scenarios where many actions co-occur, it is challenging to learn significant long-term relations, and thus these long-term features may act as noise to the model. Further experimental details are provided in Section 4.4.

3.2. Anticipation prediction Block

Inspired by [13], the module takes a feature map $F^{'} \in \mathbb{R}^{T \times D}$ and a set of anticipation queries (learnable) $LQ \in \mathbb{R}^{N_q \times D}$, as inputs. Here, N_q represents the number of queries and D is the embedding dimension, which is the same as the feature map. Action anticipation can be achieved in two different ways. The first way is to proceed directly with a transformer encoder and to learn to predict the future. An encoder sees only a glimpse of the past and learns to predict the future. On the contrary, another way is to utilize a transformer decoder. In this approach, the strength of using learnable queries with a transformer decoder is that each query learns a specific feature for a specific frame in the future. The positional encoding indicates to the transformer the order of these learnable queries and helps the model relate each query to a corresponding point in the future. Additionally, by having these learnable queries in our model, it learns to adapt to each clip, since the queries are based on the past information of each clip. Therefore, these learnable queries learn to be aware of the past. JOADAA uses these learnable queries as a link between past events and possible future ones.

$$N_q = 1 + N_f \tag{5}$$

Where in Eq. 5, 1 is for the upcoming frame that represents the ongoing action (represented in red in the Figure 2). Since we do not have access yet to this frame; thus, it is also anticipated. N_f is the number of frames to anticipate in the future to which we have no access. Information from the past, present, and future are connected by these learnable queries to improve both tasks efficiently. Later, these anticipation queries act as a pseudo-future to do the prediction of the ongoing action, see Section 3.3.

3.3. Online action prediction Block

At this stage, we feed the features of the current frame and the previously learned features of potential actions in the current time step and subsequent time steps into a decoder. Our model can classify the current frame more ac-

curately because it has pseudo-future knowledge. Modeling information this way has two effects. The prediction of the current frame is initially optimized by employing anticipation queries, and since we can access the current frame, we can also enhance the learned query on the current frame, which benefits our anticipation module. In addition, our local-to-global layers improve the performance of JOADAA. Adding a TCN layer (1D temporal convolution) helps the model capture local information. Transformers have proven to be a good tool to capture global and long-range dependencies. However, as explained earlier, this huge amount of information is not always helpful and may act as noise. Therefore, by mixing transformers with TCNs, our model learns complementary information from an updated feature map that we pass through an FC (fully connected) layer for classification. Notably, we utilize a Softmax layer for basic datasets with only one action at a time for validation and a Sigmoid layer for datasets with co-occurring actions in all categorization layers (past, future, and present).

Note that we use three different concatenation layers in our architecture. The first concatenation is between past frames features and anticipated frames features, the aim of this concatenation is to provide the decoder with a pseudo full information (past and pseudo future), which is the main idea of our paper (use AA to enhance OAD). The second concatenation is between past frames and the currently updated feature (since it is now aware of past and possible future actions). Here we only concatenate past and present because online action action detection is our main objective, which is why there is no more need for future information. The last concatenation is to use both local information learned through the TCNs and global information from the transformer decoder, which allows us to have better predictions as shown in the ablation studies Table 8.

We also use the same decoder for future frame anticipation and current frame prediction. Experiments have been conducted that showed that using different decoders does not improve the accuracy and sometimes leads to a slight decrease in accuracy. Hence, to keep the model lighter and have better prediction we keep the same weights. As for the encoders, the two of them are different; the last encoder is part of our proposed classification head, where we use a TCN to capture local dependencies and a transformer encoder to capture long-range dependencies. Therefore, our intuition was not to share the weights between the encoders as they have a separate function in our architecture.

4. Experiments

In this section, we discuss experiments carried out for online action detection and action anticipation tasks on two different types of datasets. First, we briefly describe the datasets used and explain the implementation of the experiments carried out. Second, we compare JOADAA with existing SOTA methods for both online action detection and action anticipation. Finally, we explore the effectiveness of each module of our approach by performing an ablation study. More qualitative results are provided in the supplementary materials.

4.1. Datasets

In this section, we briefly explain the datasets used in our experiments. We experiment on two types of datasets, i) sparsely annotated dataset (THUMOS'14 [15]), and ii) densely annotated datasets (Multi-THUMOS [32] and CHARADES [33]). Each of them is described below.

.**THUMOS'14**: contains 413 untrimmed videos with 20 categories of actions. The dataset is divided into two subsets: the validation set and the test set. The validation set contains 200 videos, and the test set contains 213 videos. Following common practice, we use the validation set for training and report the results in the test set. More details are available in [15].

Multi-THUMOS: contains dense, multilabel frame-level action annotations for 30 hours across 400 videos from the THUMOS'14 [15] action detection dataset. It consists of 38,690 annotations of 65 action classes, with an average of 1.5 labels per frame and 10.5 action classes per video. More details can be found in [32].

CHARADES: is composed of 9,848 videos of daily indoor activities with an average length of 30 seconds, involving interactions with 46 object classes in 15 types of indoor scenes and containing a vocabulary of 30 verbs leading to 157 action classes. Readers can find more details in [33].

4.2. Implementation Details

We implement our proposed model in PyTorch [20]. All experiments are performed on a system with 3 Nvidia V100 graphics cards. For all Transformer units, we set their number of heads to 16 and hidden units to 1024 dimensions. To learn the weights of the model, we use Adam Optimizer [18] with weight decay 5×10^{-5}. The learning rate increases linearly from zero to 5×10^{-5} in the first 40% training iterations and then decreases to zero using a cosine warm-up. Our models are optimized with a batch size of 16, and trained for 25 epochs. **Evaluation protocol**: We follow previous work and use mean average precision per frame (mAP) to evaluate performances.

4.3. Comparison with the SoTA

4.3.1 OAD Comparison on the simple dataset (THUMOS'14)

Table 1 presents the results of online action detection. For the THUMOS'14 [15] dataset we achieve state-of-the-art

	THUMOS'14	Multi-THUMOS	CHARADES
FATS [17]	59.0	-	-
IDN [8]	60.3	-	-
PKD [34]	64.5	-	-
WOAD [11]	67.1	-	-
LFB [28]	64.8	-	-
TRN [30]	62.1	39.5	18.3
PDAN [7]	62.2	32.6	16.0
MSTCT [6]	70.5	41.4	19.5
LSTR [31]	69.5	43.0	20.0
TesTra [35]	71.2	41.7	19.9
GateHUB [4]	70.7	-	-
JOADAA	**72.6**	**45.2**	**21.5**

Table 1. State of the art comparison for OAD on THUMOS'14, Multi-THUMOS, and CHARADES. Due to the lack of available OAD methods for CHARADES and Multi-THUMOS datasets, we compare also with two off-line methods PDAN and MSTCT, adapted to an online setting.

	THUMOS'14				Multi-THUMOS			CHARADES		
	1	2	4	6	2	4	6	2	4	6
TTM [27]	46.8	45.5	43.6	41.1	-	-	-	-	-	-
LSTR [31]	60.4	58.6	53.3	48.9	-	-	-	-	-	-
TesTra [35]	66.2	63.5	57.4	52.6	28.0	22.4	19.8	18.1	13.7	13.5
JOADAA	**67.7**	**63.9**	**62.9**	**59.3**	**42.5**	**37.7**	**35.2**	**20.2**	**19.5**	**19.0**

Table 2. Comparison with SOTA for the action anticipation task. 1, 2, 4, and 6 represent the number of anticipated frames. We notice that our method is more robust w.r.t. the number of anticipated frames compared to other methods where accuracy drops dramatically.

Dataset	1	2	4	6
THUMOS'14	70.5 / 67.7	71.5 / 63.9	72.2 / 62.9	72.6 / 59.3
CHARADES	20.0 / 20.7	21.4 / 20.2	21.5 / 19.5	21.4 / 19.0
Multi-THUMOS	44.5 / 42.8	45.2 / 42.5	45.0 / 37.7	45.2 / 35.2

Table 3. Effect of action anticipation prediction and online action detection using long-short-term knowledge. 1, 2, 4, and 6 are the number of anticipated frames. Best viewed in color.

Dataset	2	4	6
THUMOS'14	70.6 / 64.4	70.0 / 63.0	70.6 / 58.2
CHARADES	21.8 / 20.4	21.4 / 19.5	21.3 / 19.0
Multi-THUMOS	45.1 / 36.9	45.3 / 39.2	45.1 / 37.3

Table 4. Results of using only short-term past information on multiple datasets for online action detection and action anticipation. 2, 4, and 6 are the number of anticipated frames.

results by a margin of **1.4%**. GateHUB [4] was SoTA results for OAD on the THUMOS'14 dataset. However, they provide two results on this dataset, one with TSN as the backbone feature extractor and one with Timesformer [2]. Upon careful examination, we noticed the following

Dataset	long term past + short term past		short term past	
	LSTR	JOADAA	LSTR	JOADAA
THUMOS'14	69.5	**72.6**	65.4	**70.6**
Multi-THUMOS	42.0	**45.2**	40.0	**45.1**
CHARADES	20.0	**21.4**	19.8	**21.3**

Table 5. Comparison of JOADAA with LSTR method using long-past information. JOADAA is more robust to utilize long-past information.

points: 1) Our accuracy still surpasses theirs. 2) The GateHUB method was not compared with TesTra, which demonstrated better accuracy with the same settings. 3) GateHUB achieves SOTA results only when TimeSformer [2] is used as an RGB feature extractor, making it difficult to determine whether the results are due to the extractor or to their proposed solution. In conclusion, while the GateHUB paper argues for capturing relevant information from the past to the present, our JOADAA method, which employs a simple implementation of transformers, outperforms it along with TesTra [35].

4.3.2 OAD comparison on densely annotated datasets

We evaluate JOADAA on more complex datasets such as Multi-THUMOS [32] and CHARADES [33]. We utilize LSTR [31], TesTra [35], and TRN [30] to train on these datasets to build baseline methods, as there are no validated online methods to compare JOADAA to these datasets. JOADAA improves the baselines by **1.5%** on CHARADES [33] and **2.2%** on Multi-THUMOS [32] dataset. The main difference between our approach and baseline methods [31] and [35], is the introduction of pseudo-future knowledge to our online action prediction. It helps make more precise predictions by having a knowledge of different possible outcomes.

4.3.3 OAD comparison using off-line methods

For further comparison, we adapt offline methods to online settings. We use PDAN [7] and MSTCT [6] two SoTA methods on CHARADES and Multi-THUMOS in off-line action detection. We outperform these two methods on all three datasets THUMOS'14, Multi-THUMOS, and CHARADES.

4.3.4 AA SoTA comparison

Similarly, our model achieves SOTA results on action anticipation as noted in Table 2. When Increasing the anticipated frames from 1 to 6, TesTra's [35] accuracy drops by **13.6%** on the THUMOS'14 dataset, whereas our model decreases by only **8.4%**, which showcases robustness of our proposed solution. Also, JOADAA performs much better in more complex datasets (CHARADES and Multi-THUMOS).

In Table 3, we demonstrate how far we can foresee the future. We notice that, in general, the further we anticipate, the better the accuracy of the online action detection (blue) until it reaches a level where the accuracy stops increasing. Such a behavior makes sense because the model can learn more action dependencies by inferring more information about upcoming events. On the other hand, action anticipation results (red) decrease when the anticipation period increases, because the model has more space to explore.

4.4. Ablation study

In this section, we discuss how the different modules contribute to JOADAA.

4.4.1 Ablation on the past processing block

First, we analyze the use of long-range past features on different datasets. As discussed in Section 3, past information can be used in two manners, either using only short-term past (32 frames) or long-short-term past (512+32 frames). This past information is used to infer the pseudo-future in our approach. In Tables 4 and 5, we observe that our model is more robust when it comes to using only short-term past information (decreases by **2%**) on the THUMOS'14 [15], unlike LSTR [31] where the accuracy decreases by **4.1%**. One important result of our study is that long-past knowledge is more important for simple actions (single-action datasets) than for complex actions (densely annotated datasets). This is because numerous actions may occur simultaneously without being connected in densely annotated datasets, making it more challenging to infer relations from them. As a result, including information from the distant past can skew model predictions.

Module	THUMOS'14
Transformer encoder	71.5
LSTM+Conv	54.2

Table 6. Comparing two techniques for past information processing. We use a transformer encoder and a set of LSTM blocks with a convolution layer.

Recently, transformers have been widely used, since they outperformed the existing approaches such as 3D-CNNs and RNNs. In fact, 3D-CNNs are known to be good general feature extractors as they can capture overall visual appearances in a video. However, their CNN filters capture pixel-level information in a local neighborhood but struggle with long-term dependencies. Therefore, we limit the use of 3D-CNNs to extract video clip features for our architecture. Furthermore, action detection tasks require a strong grasp of long-range temporal dependencies, and transformers excel at capturing long-term information compared to RNNs. Therefore, the transformers are the best choice for OAD and AA tasks. However, most papers lately use transformers based on the previous intuition without any justification.

Table 6 presents a comparison study between RNNs (LSTMs [25]) and transformers. We replace our first encoder for past information processing with 3 blocks of LSTM and a convolution layer to reduce the feature map size. Results show that transformers are better suited for capturing long-range dependencies and produce far more better results which justifies our design choice.

4.4.2 Ablation on the action anticipation module

Another ablation study is done in Table 7. We conduct two main experiments: one with the full JOADAA model and the other one without the Action Anticipation (AA) module. We can see that the AA module enhances online action detection, which supports our claim that combining AA and OAD leads to better results.

Dataset	OAD+AA	OAD
THUMOS'14	72.6	71.2

Table 7. Analyzing the JOADAA behavior with and without action anticipation.

4.4.3 Ablation on the OAD prediction layer

Dataset	TCN+TR. Encoder	FC
THUMOS'14	72.6	69.7

Table 8. Effect of fusing local and global information on OAD. FC stands for fully-connected layer. As expected capturing different type of dependencies provides better results.

Table 8 shows the effect of fusing local and global knowledge, in contrast to using directly the output of the decoder on the current frame which carries only global information in it. By doing so, our results increase by **2.9%**. As argued earlier, this is due to the fact that TCNs can extract local changes and better detect relations in neighboring frames, whereas baseline transformers capture long-range dependencies that sometimes are not adapted to predicting the current frame events.

4.5. Qualitative Analysis

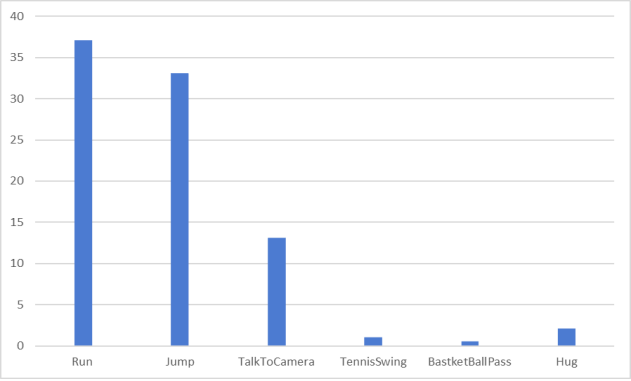

Figure 3. Action anticipation accuracy improvement on six actions w.r.t. TesTra model. This is performed on the Multi-THUMOS dataset, using 4 frames as anticipation length.

In this section, we analyze the effectiveness of our method on densely annotated datasets. We study anticipation improvement on six different actions, from the Multi-THUMOS dataset, according to their complexity as shown in Figure 3. We observe that the gain in some of these actions can reach 37%, while in some other actions, it is almost zero.

In fact, our prediction block anticipates the upcoming frame alongside future frames. By having access to the current frame our model can correlate the anticipated action to the real action, hence we can learn to better anticipate the current frame, leading to a better-performing anticipation module.

Upon closer examination of these actions, we find that the improvement is particularly important for activities where there are multiple dependencies, or if the activity is interconnected with many other actions. The action **Run** for instance, has correlations with up to seven other activities, as illustrated in Figure 1.

The qualitative results in Figure 3 demonstrate the robustness of JOADAA for complex correlated activities. This opens doors for future studies to analyze OAD and action anticipation on complex dense datasets.

5. Conclusion

Online action detection and anticipation are important fields in computer vision that have many real-world applications. These two tasks are highly correlated, and that is why we design JOADAA to address both tasks jointly, improving one using the other and vice versa. Furthermore, we discuss the limitations of OAD and action anticipation for sparsely and densely annotated datasets.

Our model is limited in terms of effectively using long-range past features, especially for densely annotated datasets. Past knowledge undoubtedly adds to current knowledge and should lead to improvements. However, as demonstrated in this study, just adding pre-extracted features to transformers can also introduce noise. In the future, we are interested in tackling this limitation by modeling past features more effectively. One possible solution is to use an intermediate filter to learn only important features [5] or to learn the dependencies using a graph model to model only relevant features following [14].

Acknowledgments

The COFUND BoostUrCAreer program has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Curie grant agreement No 847581, from the Région SUD Provence-Alpes-Côte d'Azur and IDEX UCAjedi. This work has also been supported by the French government, through the ACTIVIS project managed by the National Research Agency (ANR) with the reference number ANR-19-CE19-0004.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 1

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 6

[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. 2, 3

[4] Junwen Chen, Gaurav Mittal, Ye Yu, Yu Kong, and Mei Chen. Gatehub: Gated history unit with background suppression for online action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19925–19934, 2022. 6

[5] Rui Dai, Srijan Das, and Francois Bremond. Ctrn: Class-temporal relational network for action detection. *arXiv preprint arXiv:2110.13473*, 2021. 8

[6] Rui Dai, Srijan Das, Kumara Kahatapitiya, Michael S Ryoo, and François Brémond. Ms-tct: multi-scale temporal convtransformer for action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20051, 2022. 6, 7

[7] 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/CVF Winter Conference on Applications of Computer Vision*, pages 2970–2979, 2021. 3, 6, 7

[8] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Learning to discriminate information for online action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 809–818, 2020. 3, 6

[9] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. *arXiv preprint arXiv:1707.04818*, 2017. 3

[10] Mingfei Gao, Mingze Xu, Larry S Davis, Richard Socher, and Caiming Xiong. Startnet: Online detection of action start in untrimmed videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5542–5551, 2019. 3

[11] Mingfei Gao, Yingbo Zhou, Ran Xu, Richard Socher, and Caiming Xiong. Woad: Weakly supervised online action detection in untrimmed videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1915–1923, 2021. 3, 6

[12] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13505–13515, 2021. 3

[13] Dayoung Gong, Joonseok Lee, Manjin Kim, Seong Jong Ha, and Minsu Cho. Future transformer for long-term action anticipation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3052–3061, 2022. 1, 3, 4

[14] Mohammed Guermal, Rui Dai, and François Brémond. Thorn: Temporal human-object relation network for action recognition. *arXiv preprint arXiv:2204.09468*, 2022. 8

[15] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. 5, 7

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

[17] Young Hwi Kim, Seonghyeon Nam, and Seon Joo Kim. Temporally smooth online action detection using cycle-consistent future anticipation. *Pattern Recognition*, 116:107954, 2021. 6

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 1

[20] Automatic Differentiation In Pytorch. Pytorch, 2018. 5

[21] Sanqing Qu, Guang Chen, Dan Xu, Jinhu Dong, Fan Lu, and Alois Knoll. Lap-net: Adaptive features sampling via learning action progression for online action detection. *arXiv preprint arXiv:2011.07915*, 2020. 3

[22] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017. 3

[23] Zheng Shou, Junting Pan, Jonathan Chan, Kazuyuki Miyazawa, Hassan Mansour, Anthony Vetro, Xavier Giroi Nieto, and Shih-Fu Chang. Online action detection in untrimmed, streaming videos-modeling and evaluation. In *ECCV*, volume 1, page 5, 2018. 3

[24] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016. 3

[25] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019. 7

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 3

[27] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7565–7575, 2021. 6

[28] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term

feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019. 6

[29] 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. 3

[30] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S Davis, and David J Crandall. Temporal recurrent networks for online action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5532–5541, 2019. 6, 7

[31] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, 34:1086–1099, 2021. 1, 3, 6, 7

[32] 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, 3, 5, 7

[33] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *IEEE Transactions on Image Processing*, 29:5491–5506, 2020. 5, 7

[34] Peisen Zhao, Lingxi Xie, Ya Zhang, Yanfeng Wang, and Qi Tian. Privileged knowledge distillation for online action detection. *arXiv preprint arXiv:2011.09158*, 2020. 3, 6

[35] Yue Zhao and Philipp Krähenbühl. Real-time online video detection with temporal smoothing transformers. In *European Conference on Computer Vision*, pages 485–502. Springer, 2022. 1, 3, 6, 7

[36] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 3