Unintentional Action Localization via Counterfactual Examples

Jinglin Xu, Guangyi Chen, Jiwen Lu, *Senior Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

***Abstract*— How do humans localize unintentional action like "*A boy falls down while playing skateboard*"? Cognitive science shows that an 18-month-old baby understands the intention by observing the actions and comparing the feedback. Motivated by this evidence, we propose a causal inference approach that constructs a video pool containing intentional knowledge, conducts the counterfactual intervention to observe intentional action, and compares the unintentional action with intentional action to achieve localization. Specifically, we first build a video pool, where each video contains the same action content as an original unintentional action video. Then we conduct the counterfactual intervention to generate counterfactual examples. We further maximize the difference between the predictions of factual unintentional action and counterfactual intentional action to train the model. By disentangling the effects of different clues on the model prediction, we encourage the model to highlight the intention clue and alleviate the negative effect brought by the training bias of the action content clue. We evaluate our approach on a public unintentional action dataset and achieve consistent improvements on both unintentional action recognition and localization tasks.**

***Index Terms*— Temporal action localization, causal inference, video pool, counterfactual intervention, intention.**

I. Introduction

CONSIDERING a video such as "*A boy falls down while playing skateboard*" shown in Fig. 1, humans can easily understand the intention [1], i.e., "playing skateboard" is an intentional action and "falling down" is an unintentional action. For intelligent systems, such as self-driving vehicles and robots, understanding the intention behind observed action is of critical importance to avoid risks and failures. Existing video understanding methods can answer what the action content is (i.e., action recognition [2]–[7]) or when the action starts and ends (i.e., temporal action localization/ detection [8]–[12]), and how the action quality is (i.e., action quality assessment [13]–[17]), yet cannot explain the reason why the action fails (i.e., the action transition from intentional to unintentional).

It is challenging for a machine to localize unintentional action only by observing failures in the video since it cannot imagine what the intentional development of unintentional action should be. Current unintentional action localization (UAL) methods, e.g., [18], recognize action intentionality by using spatial-temporal features in a likelihood manner. However, most of these methods may ignore the fact that video features contain both action content and intention clues, where the former may bring the training bias which misleads the model to learn spurious correlations instead of real causations. Specifically, in the public unintentional action dataset [18], the action content brings the training bias, that is to say, the time-stamps of unintentional actions occurring are biased for different action contents. For example, in the training set, the time-stamps of occurring unintentional actions in the "Playing skateboard" videos are different from the time-stamps of occurring unintentional actions in the "Doing handstand" videos. Therefore, the model prediction may be falsely attributed to the action content instead of occurring unintentional action. This is the spurious correlation brought by the training bias. Besides, these spurious correlations learned from the training bias are difficult to be transferred to the testing. For instance, the time-stamps of occurring unintentional actions in the "Playing skateboard" videos are concentrated in different time segments respectively in the training and testing sets. An intuitive solution is to disentangle video features into the action content and intention clues to remove the bias. Unfortunately, it needs a large number of fine-grained annotations of action contents and a more complex disentangled model.

To address this problem, we propose an Unintentional Action Localization via Counterfactual Examples (UAL-CE) approach to disentangle the effects of content and intention clues, which mitigates the negative effect brought by biased action content and highlights the causal effect of intention on model prediction. UAL-CE consists of two stages: 1) conducting the counterfactual intervention to observe the intentional development, and 2) comparing original unintentional action with counterfactual intentional action, simulating the observing and comparing processes of baby understanding the intention. To this end, we first construct a causal graph, where action content is a confounder that causes the spurious correlation between intention and prediction. Then we construct a video pool containing intentional knowledge and conduct the counterfactual intervention to imagine the intentional situation of unintentional action, which cuts off the dependence of intention on action content. Given an original video composed of the former intentional action and

Manuscript received July 18, 2021; revised January 22, 2022 and March 19, 2022; accepted March 30, 2022. Date of publication April 15, 2022; date of current version April 27, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62106124, Grant 62125603, and Grant U1813218; in part by the China Postdoctoral Science Foundation under Grant 2020M680564; and in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Clinton Fookes. *(Jinglin Xu and Guangyi Chen contributed equally to this work.) (Corresponding author: Jiwen Lu.)*

The authors are with the Beijing National Research Center for Information Science and Technology (BNRist) and the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: xujinglinlove@gmail.com; guangyichen1994@gmail.com; lujiwen@tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TIP.2022.3166278

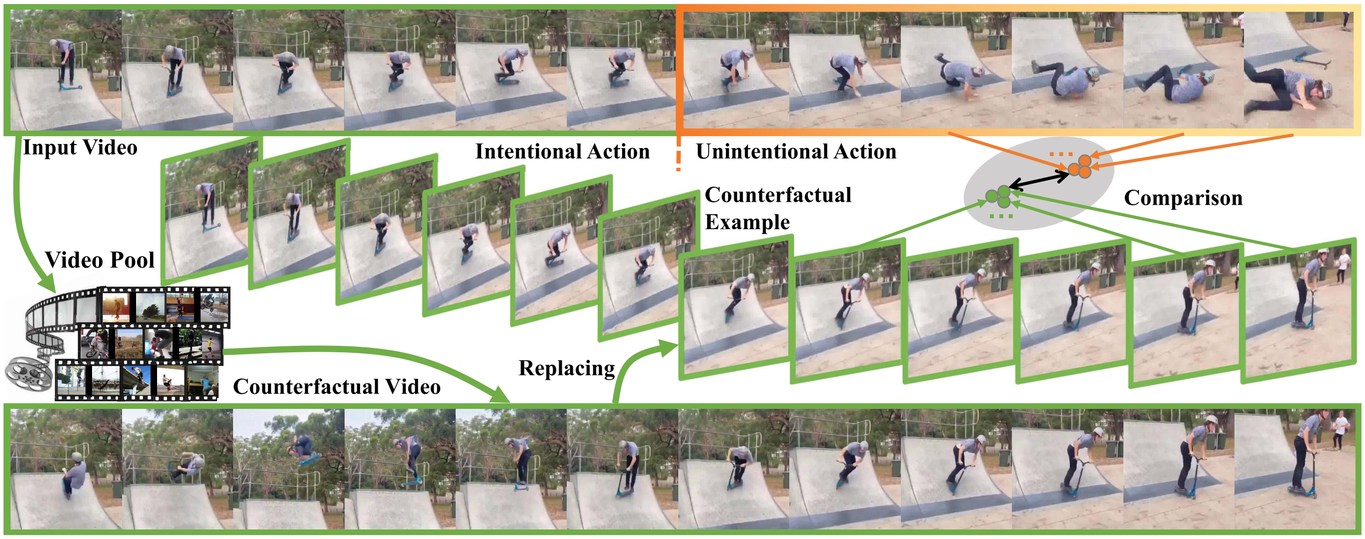


Fig. 1. How do humans localize unintentional action like "*A boy falls down while playing skateboard?*" We argue that it requires (1) *observing* how the action performs intentionally and (2) *comparing* unintentional and intentional actions. We propose a causal inference approach that constructs a video pool containing intentional knowledge, uses counterfactual intervention to generate the counterfactual example, and makes the comparison between factual unintentional video and counterfactual intentional video to analyze the causation of intention.

subsequent unintentional action, we utilize the former intentional action to search for the intentional action in the counterfactual video, and concatenate the former intentional action in the original video and subsequent intentional development in the counterfactual video to generate the counterfactual example, where the counterfactual video is composed of the intentional action and subsequent intentional development. Finally, we train the model by maximizing the difference between the predictions of original and counterfactual examples, which encourages the model to detect the action transition from intentional to unintentional, i.e., finding out the reason why the action fails. We alleviate the negative effect of action content by subtracting the counterfactual prediction from the original one. We evaluate UAL-CE on the OOPS dataset for unintentional action recognition and localization tasks and achieve consistent improvements.

The main contributions are summarized as follows:

- We introduce the causal inference into UAL to analyze the causation of intention and alleviate the spurious correlations of biased action content.
- We construct a video pool with intentional knowledge to equip the machine with common sense to know what happens if humans change the original fact and do the thing successfully instead.
- We propose the counterfactual intervention to observe how actions perform intentionally and generate counterfactual examples for calculating causal effects.
- Our approach significantly outperforms other state-of-the-art methods on the OOPS dataset for the tasks of unintentional action recognition and localization.

II. Related Work

In this section, we briefly review four related topics, including temporal action localization, unintentional action localization, anomaly detection, and causal inference.

A. Temporal Action Localization

Temporal action localization, different from action recognition [19]–[21], which not only recognizes the action category but also localizes the start and end of the action. In early pioneering work, Gan *et al.* [22] proposed DevNet that was the first to simultaneously detect pre-defined events and provide key spatial-temporal evidence. Later, many deep learning-based methods [8], [10], [11], [22]–[27] utilized temporal proposals to localize human action. For example, Shou *et al.* [8] proposed the multi-stage CNNs by combining three segment-based 3D ConvNets, including a proposal network, a classification network, and a localization network. Chen *et al.* [25] proposed a relation attention module to effectively exploit the relation between video proposals, which can be simply applied to various action localization algorithms. Zeng *et al.* [26] proposed Graph Convolutional Networks (GCNs) over the graph to model the relations among different proposals and learn powerful representations for the action localization. To solve the problem of heavy tuning of locations and sizes corresponding to different anchors, Lin *et al.* [28] proposed an efficient and effective anchor-free temporal localization method. In addition, some other related works were based on the probability distribution curve [9], [29]–[31]. For instance, Shou *et al.* [9] proposed a convolutional de-convolutional filter to simultaneously perform spatial downsampling and temporal upsampling, and design a network to predict actions at frame-level. Yang *et al.* [29] presented temporal preservation convolutional (TPC) network equipped with 3D ConvNets with TPC filters, which preserves all the temporal information to make frame-level action predictions. However, temporal action localization only tells us the action category and its temporal boundary but cannot recognize the action intentionality variation to explain the reason why the action fails.

B. Unintentional Action Localization

Unintentional action localization aims at understanding action intention and localizing when an intentional action turns into unintentional action. To understand the action intention, Epstein *et al.* [18] collected an unintentional action video dataset (i.e., OOPS) and trained a three-way classifier to recognize the action as intentional, transitional, and unintentional. Furthermore, Epstein and Vondrick [32] further annotated the goals of original video actions to improve supervision quality and trained discriminative video representations. The inputs of these models are the spatial-temporal features containing both action content and intention clues, which usually misleads the model prediction with the spurious correlation brought by biased action content. Hence, we propose a causal inference approach to mitigate the negative effect of training bias.

C. Anomaly Detection

Anomaly detection aims to determine all dissimilar instances due to several reasons, such as malicious actions, system failures, intentional fraud, violence, or aggression [33]–[36]. In recent years, deep learning-based methods employed deep generic knowledge [37], stacked recurrent neural network [38], cascaded deep network [39] and plug-and-play CNNs [40], [41] for anomaly detection. Since real-world anomalous events are complicated and diverse, it is difficult to list all of the possible anomalous events. Therefore, the solution of some specific anomalous events cannot be generalized to detect other anomalous events, which is the most challenging problem for anomaly detection. Different from anomaly detection, the difficulty of UAL is to learn the knowledge of human action intention and localize unintentional action when only observing the video containing unintentional failures. Anomaly detection focuses on abnormal behavior patterns that are intentional, e.g., “retrograde”, “fight”, and “steal”, while UAL focuses on unintentional failures, e.g., “fall down” and “slip off”.

D. Causal Inference

Causal inference [42]–[44] empowers the ability to pursue the causal effect, which investigates the subsequent effect when the cause is changed. Understanding the causations is critical to remove data bias [45]–[48], build transparent and explainable model [49]–[52], promote fairness [53]–[55], and recover from missing data [56], [57]. Recently, causal inference has been applied for different fields including natural language processing [58], [59], reinforcement learning [60], [61] and computer vision [48], [62], [63]. Inspired by causal effects, we introduce causal inference to conduct the counterfactual intervention to alleviate the negative effect brought by the training bias of action content. Different from recent debiasing methods used for visual question answering [64], [65] and multi-label image classification [52], [66], our approach constructs extra data containing intentional knowledge and introduces it into the model training to remove the spurious correlations of biased action content clues and highlight the intention clues in the videos.

III. Approach

In this section, we will introduce how to localize the unintentional action via counterfactual examples. We first formally define the problem and build a causal graph of UAL to analyze the causations. Then we present our causal inference approach, including a video pool construction, counterfactual intervention, and ETT Calculation, and describe the details of network architecture and optimization method.

A. Problem Definition

The intention is of critical importance for personal decision-making, where the intentional and unintentional actions serve as two opposite situations. Existing method [18] defines a task that localizes the time-stamps of unintentional actions occurring in realistic videos to teach the model to understand action intention. We formulate this task as a sequential recognition problem, which extracts the clues from prior information and current observation to recognize whether the unintentional action occurs at the current time-stamp. Given a video including T frames $X = \{\boldsymbol{x}_t | t = 1, 2, \cdots, T\}$, the model recognizes the intentionality of action in each frame as one of three action state categories $Y = \{y_t | t = 1, 2, \cdots, T\}$, where $y_t \in \{0, 1, 2\}$ denote different states of the action including *intentional*, *transitional*, and *unintentional*. Predicting the action state of each video frame, we localize the unintentional action by selecting the time that most likely occurring unintentional action, which is implemented by localizing the transition from intentional to unintentional:

$$\hat{t} = \arg\max_{t} P(y_t = 1 | \boldsymbol{x}_t), \tag{1}$$

where $P(y_t = 1|\boldsymbol{x}_t)$ denotes the probability of transition from intentional to unintentional at this time, indicating unintentional action beginning (transition to unintentional action). In our approach, suppose that a video containing unintentional action can be divided into two parts, including an intentional part C and the combination of transitional and unintentional parts U. The former C denotes the action content like “playing skateboard”, while the latter U consists of the reason why the action fails and the time of unintentional action occurring. It is difficult to overcome the data bias by using U and removing C since the above two parts U and C of a video are coupled.

B. Causal Graph Construction

In Fig. 2, we construct a causal graph of UAL where the nodes include an input video X, action content C, unintentional action U, and intentionality prediction Y. Considering an example that “*A boy falls down while playing skateboard*”, the nodes are described as follows:

- X denotes the input video which contains the action content clue such as “playing skateboard” and the intention clue like “falls down”.
- C is the action content such as “playing skateboard”.
- U is the unintentional action that denotes the failure occurring in the input video such as “falls down”.
- Y is the prediction of the model, which denotes the time-stamp of unintentional action occurring.

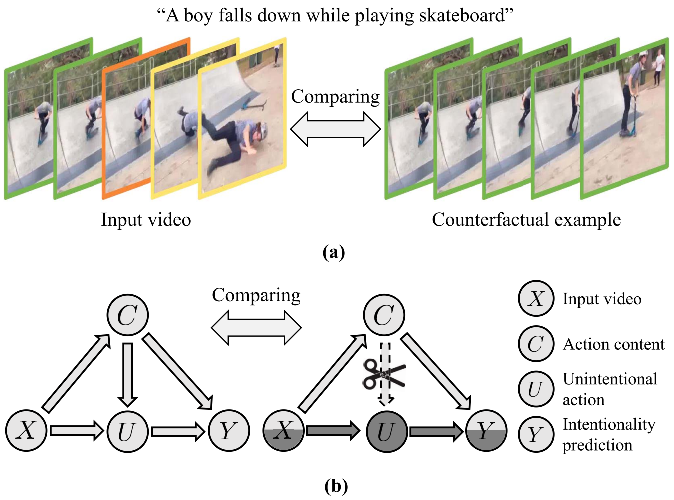


Fig. 2. The causal inference approach of unintentional action localization. (a) shows input unintentional video and counterfactual intentional video of "playing skateboard". (b) illustrates the calculation of the effect of treatment on the treated (ETT) and the corresponding operations on the causal graph of UAL. In (b), the left and the right parts respectively denote the causal graph before and after using the counterfactual intervention, where the shadow indicates that the variable is counterfactual.

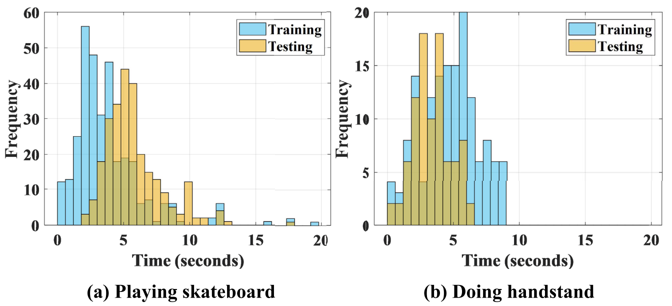


Fig. 3. Examples of statistical training bias on the OOPS dataset. We show the frequencies of unintentional actions occurring at different time-stamps on both training and testing sets for two action contents including (a) "Playing skateboard" and (b) "Doing handstand". The bias exists in different action contents and cannot be transferred from the training to the testing sets. Best viewed in color.

The link means the causal dependence between two variables, where $U \rightarrow Y$ denotes that U is the cause and Y is the effect. The links in our structural causal model shown in Fig. 2 are described as follows:

- $U \leftarrow X \rightarrow C$ shows that a video with unintentional action contains both action content and intention clues.
- $U \rightarrow Y \leftarrow C$ shows that both action content and intention clues have influences on the final intentionality prediction, where $U \rightarrow Y$ is the real causation of intention and $C \rightarrow Y$ is the spurious correlation brought by the training bias. As shown in Fig. 3, in the training set, unintentional actions occur in $[1.6, 4.2]$ seconds in the "Playing skateboard" videos and those occur in $[2.1, 6.3]$ seconds in the "Doing handstand" videos. It is obvious that the time-stamps of unintentional actions occurring are biased for different action contents (i.e., "Playing skateboard" and "Doing handstand").
- $U \leftarrow C \rightarrow Y$ denotes that the action content C has influences on both the unintentional action U and the final prediction Y. C is defined as a confounder [67] which confounds the real causation between action intention and intentionality prediction. As shown in Fig. 3, the real cause of model prediction may be falsely attributed to the action content instead of unintentional action occurring.

C. Localization With Counterfactual Examples

Although strong deep networks can fit visual features with intentionality labels in a correlation-driven manner, the real causations of the model prediction are still nontransparent. The visual features contain action content and intention clues, where the former usually brings the training bias which misleads the model to learn the spurious correlations instead of real causations. Disentangling the action content and intention clues in the features needs a large number of fine-grained annotations and a more complex disentangled model. Hence, we propose UAL-CE to disentangle the effects of different clues on the model prediction and mitigate the spurious correlations. Motivated by a baby understanding the intention via observing the actions and comparing the feedbacks, UAL-CE contains two steps: (1) conducting the counterfactual intervention to imagine and observe the intentional development of original unintentional action and (2) comparing original unintentional action with counterfactual intentional action to analyze the intention. UAL-CE can mitigate the negative effect brought by the training bias of action content and highlight the causal effect of intention on model prediction.

UAL-CE is based on the counterfactual intervention in the causal inference theory [42], [50], [68] to mitigate the negative effect of the biased data. First, we show that the construction of the counterfactual example is beneficial to learning the de-confounded representations. The construction process is similar to the intervention process, which fixes the value of one variable and observes the change of other variables. According to the causal inference theory in [68], the intervention takes the form of fixing the value of a variable in the graphical model, which amounts to performing a kind of surgery on the graphical model (i.e., removing all edges directed into that variable). When using the counterfactual example, we conduct the intervention on variable U and obtain the prediction as $P(Y|do(U = u))$, where the do-operator indicates we intervene to make $U = u$. In the causal theory, when $P(Y|U = u) \neq P(Y|do(U = u))$, variables U and Y are confounded by the confounder C. We block the effect from C to U and obtain the joint effect of both U and C on Y as $P(Y|C, do(U = u))$. To mitigate the biased effect from the confounder C, we keep it invariant, change variable U from factual to counterfactual, and calculate the prediction difference, which is defined as the Effect of Treatment on the Treated (ETT) [68]. This is equivalent to using the Randomized Control Trial [69] of medical applications in the model training process, where the effect of the factual one is the experimental group and the counterfactual one is equal to the control group. By comparing the control group and experimental group, we can mitigate the effect of the environmental variable C and obtain unbiased results. In the following subsections, we will introduce how to conduct the counterfactual intervention based on a video pool and optimize the model by maximizing the ETT.

1) Video Pool Construction: Although humans have the ability to easily imagine the counterfactual situation, it is difficult for a machine to equip with such reasoning ability due to the lack of common sense. Therefore, we build a video pool, called OOPS-CE, containing intentional knowledge to imagine and observe what happens if humans change the original fact and do the thing successfully instead. OOPS-CE, denoted as $\mathcal{V}$, contains more than 3000 videos downloaded from YouTube. The number of downloaded videos is equal to that of original annotated videos (containing available keywords) for training, which is one-by-one correspondence via the keywords of action content.

- **Collection.** We search videos from YouTube with the keywords of action content provided by the OOPS [18] dataset, such as "playing skateboard", "doing handstand", "ride a unicycle", and "playing the trampoline", and download the top one video in searched results for each keyword. The above keywords come from the annotation files (i.e., "train.josn" and "val.josn") of the OOPS dataset which can be found in "Natural language descriptions.zip" downloaded at the official website https://oops.cs.columbia.edu/data/. Fig. 4 displays some pairwise videos from the OOPS dataset and video pool (OOPS-CE), from left to right including "Riding a unicycle", "Riding over obstacles", "Rope balance", "Surfing", "Pole dance", "High jump", "Workout in gym", and "Jumping over another". For example, the top-left corner video pair shows a man riding a unicycle falling off or not and the bottom-right shows a man jumping over a person or not, where the top line of each pair of videos denotes the original video and the bottom is the video from $\mathcal{V}$. OOPS-CE is available at https://github.com/xujinglin/UAL-CE.
- **Quality Control.** During the process of building OOPS-CE, the ambiguities of some keywords may lead to downloaded videos with different action content from the original labeled training videos. Therefore, we repeat the download three times to verify the reliability of the downloaded videos and further ask three workers to check and update them to control the quality of the video pool. Fig. 5 shows two examples filtered out when controlling the quality of OOPS-CE, where the top line of each video pair denotes the original video and the bottom line denotes the filtered out video. It can be seen that the action contents of filtered-out videos are irrelevant to original labeled training videos, despite they are described by the same keywords. For instance, the video pair in Fig. 5 (a) have the same keyword "Uneven bars" but contain different action content, since the performers use "Uneven bars" in different manners to do different things. Besides, the video pair in Fig. 5 (b) contain the same keyword "Sleeping at the desk" while their action contents are different because the performers are a person and a cat, respectively.
- **Statistics.** The video pool (i.e., OOPS-CE) applied to train contains 3000+ videos since only 3000+ original labeled training videos in the OOPS dataset have available keywords for searching and the rest training videos are described by "`Don't know!`" which cannot be used to construct counterfactual data. Fig. 6 shows the statistics of both the OOPS dataset and OOPS-CE, including the distributions of original video lengths, action class distribution, and scene class distribution. It can be seen that they are independent and identically distributed, demonstrating the videos from the OOPS dataset and OOPS-CE are in the same knowledge domain. Actually, the videos in the video pool (i.e., OOPS-CE) $\mathcal{V}$ filmed from the real world ensure the same action content as original videos but are difficult for keeping consistent with the original video completely, since a large number of "fail" videos cannot be reproduced. Therefore, we keep the scene classes of each pair of videos as same as possible during the process of building OOPS-CE, as shown in Fig. 6 (c). Besides, to better distinguish action content categories (such as motorcycling, somersaulting, and driving a car) and action state categories (intentional, transitional, and unintentional), we show the statistics of action state categories for all video frames in the OOPS dataset in Fig. 7.

2) Counterfactual Intervention: Although an off-line video pool $\mathcal{V}$ has been built to represent the common sense that the things are successfully performed. The videos from $\mathcal{V}$ cannot be directly used as intentional actions since these untrimmed videos contain much noisy information, which cannot generate reliable counterfactual examples and cannot make the constructed video pairs comparable. Hence, we propose to align the intentional action for pairwise videos from the video pool and OOPS dataset. The alignment is beneficial to generate a counterfactual example by selecting an appropriate part and does not affected by the background content. Given an original labeled training video consisting of the former intentional action $C = \boldsymbol{c}$ and subsequent unintentional action $U = \boldsymbol{\mu}$, the alignment process is formulated as:

$$\boldsymbol{v}_c = \mathop{\arg\min}_{\boldsymbol{v} \in V} \mathcal{D}(\boldsymbol{v}, \boldsymbol{c}), \tag{2}$$

where the video V from $\mathcal{V}$ with the same action content as $\boldsymbol{c}$ denotes the counterfactual video, under a basic assumption that the videos from $\mathcal{V}$ are intentional. $\mathcal{D}(\boldsymbol{v}, \boldsymbol{c})$ calculates the ℓ_2 distance between a video clip $\boldsymbol{v}$ from the video V and the former intentional action $\boldsymbol{c}$ in the original video. The alignment-based method calculates $\mathcal{D}(\boldsymbol{v}, \boldsymbol{c})$ for all clips in video V and outputs the video clip $\boldsymbol{v}_c$ that makes $\mathcal{D}(\boldsymbol{v}, \boldsymbol{c})$ reach the minimum, i.e., $\mathcal{D}(\boldsymbol{v}_c, \boldsymbol{c})$, where $\boldsymbol{v}_c$ denotes the searched video clip in the counterfactual video V. This alignment process is unsupervised, which does not use any extra annotation. After aligning $\boldsymbol{v}_c$ with $\boldsymbol{c}$, we slice the counterfactual video V into two parts including the aligned intentional action $\boldsymbol{c}_c$ and subsequent intentional development $\boldsymbol{\mu}_c$. The $\boldsymbol{c}_c$ is composed of $\boldsymbol{v}_c$ and the segment before $\boldsymbol{v}_c$. Finally, the counterfactual example is generated by concatenating the former intentional action $\boldsymbol{c}$ in the original unintentional video and subsequent intentional development $\boldsymbol{\mu}_c$ in the counterfactual video V, where the lengths of counterfactual example

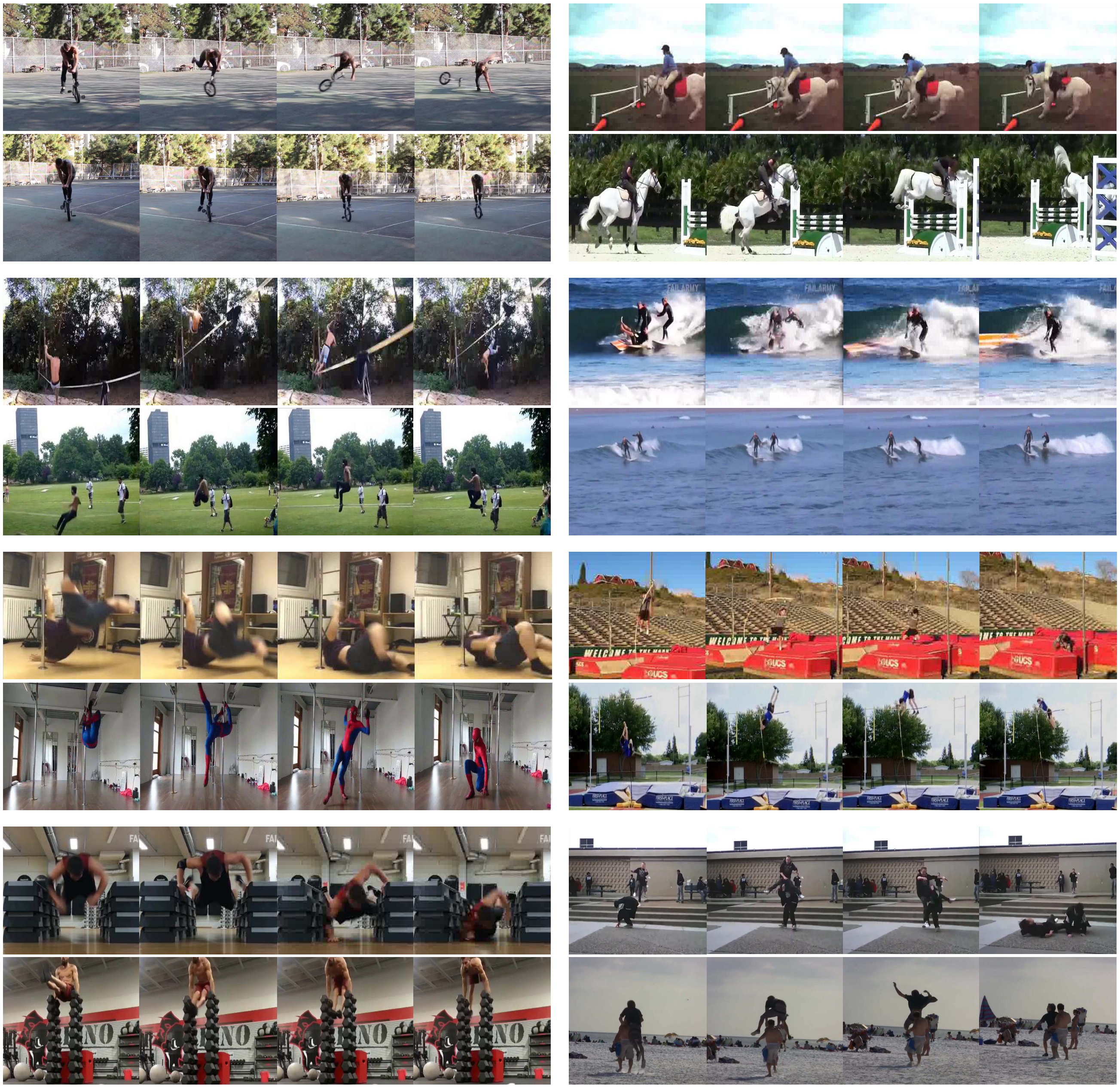


Fig. 4. Each pair of videos shows an example of factual unintentional action in the OOPS dataset and counterfactual intentional action in the video pool (i.e., OOPS-CE). By crawling publicly available "success" videos from the web, we create a video pool containing intentional knowledge, corresponding to "fail" videos one by one. We show several video pairs, such as "riding a unicycle," "riding over obstacles," "rope balance," "surfing," "pole dance," "high jump," "workout in gym," and "jumping over another." Taking the top-left corner video pair as an example, the top line of this video pair shows a man riding a unicycle falling off and the bottom line displays the man doing this thing successfully.

and original factual video are fixed into the same size via downsampling or upsampling. Note that only the original labeled training videos owning counterfactual examples are applied for causal inference during the training.

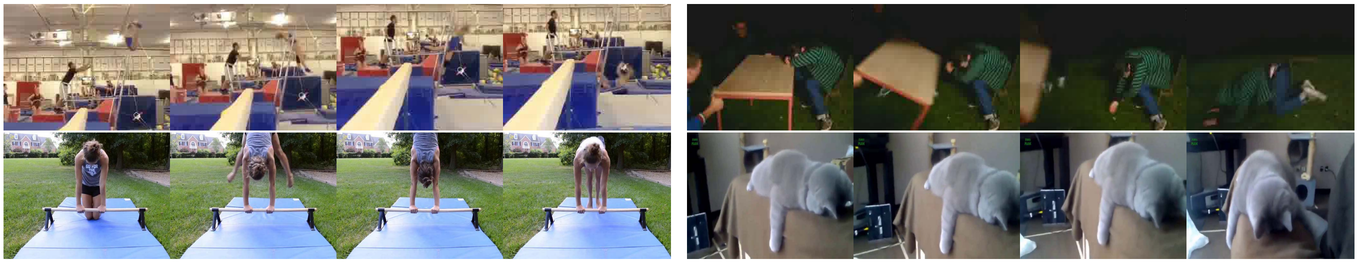
3) ETT Calculation: The Effect of Treatment on the Treated (ETT) denotes the real effect when the treatment is applied [42]. In our approach, the occurrence of the unintentional action is regarded as the treatment. Then we learn the effect of treatment (unintentional action) by comparing the original unintentional video and the counterfactual example. As shown in Fig. 2, we calculate the ETT by subtracting the counterfactual prediction from the original prediction. Specifically, the original likelihood prediction can be formulated as:

$$Y_{U=\boldsymbol{\mu}} = P(Y|X(C, U)), \tag{3}$$

where $X(C, U)$ denotes that the input video X contains both action content C and intention U clues. The counterfactual prediction can be written as:

$$Y_{U=\boldsymbol{\mu}_c} = P(Y|X(C, do(U = \boldsymbol{\mu}_c))), \tag{4}$$

where $U = \boldsymbol{\mu}_c$ denotes the counterfactual situation, such as "intentionally playing skateboard successfully". As shown in



(a) Uneven bars

(b) Sleeping at the desk

Fig. 5. Some examples are filtered out when controlling the quality of video pool (i.e., OOPS-CE). In (a), the video pair is described by the same keyword “uneven bars” but contains different action content, since the performers do different things. In (b), the video pair has the same keyword “sleeping at the desk” while their action contents are different because the performers are a person and a cat, respectively.

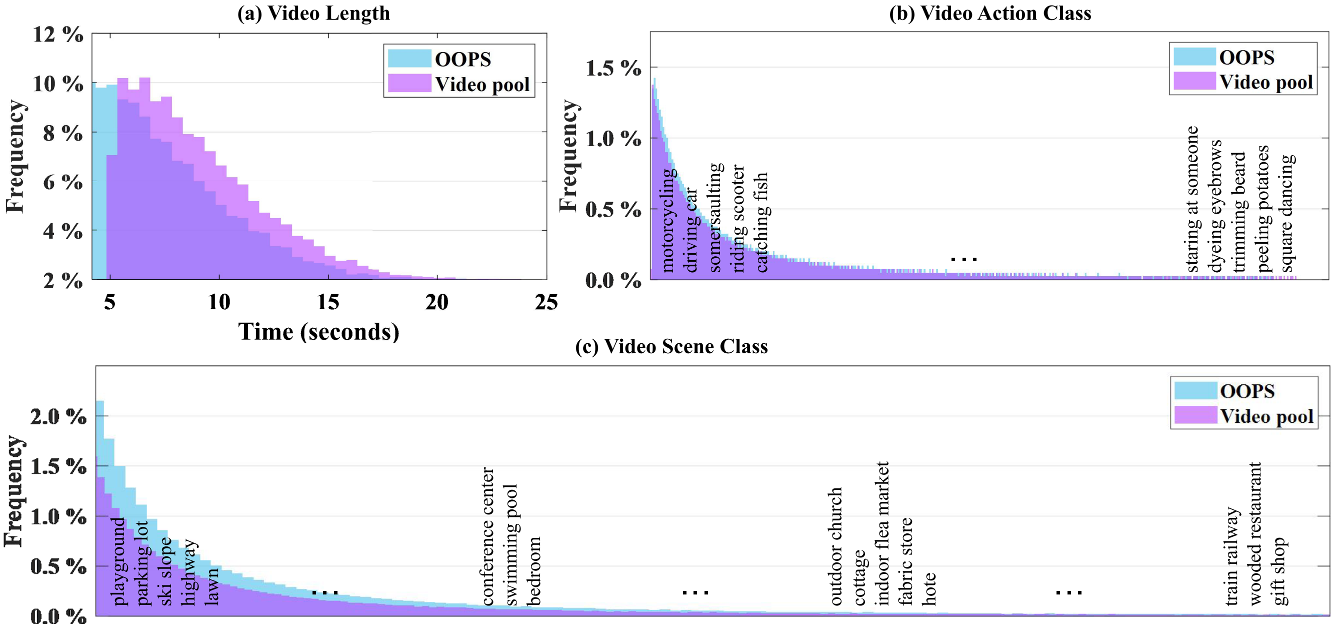


Fig. 6. The statistics of the OOPS dataset and video pool (i.e., OOPS-CE). In (a), the video length distribution of the OOPS dataset is slightly different from that of the video pool, which does not affect the causal inference during the training. (b) indicates the OOPS dataset and video pool contain the same action content, where two action class distributions almost overlap. (c) denotes the video scene class distributions of the OOPS dataset and video pool are slightly different since the “failure” in some scenes cannot be reproduced.

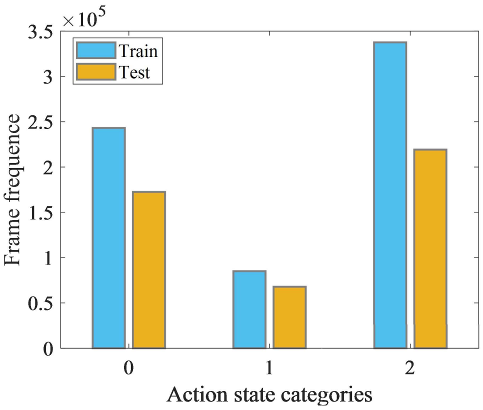


Fig. 7. The statistics of action state categories for all video frames in the OOPS dataset. The x-axis indicates action state categories, where 0, 1, and 2 indicate intentional action, transition to unintentional action, and unintentional action. The y-axis denotes frame frequency for all videos in OOPS.

Fig. 2 (b), the calculation of ETT is written as:

$$ETT = \mathbb{E}[Y_{U=\boldsymbol{\mu}} - Y_{U=\boldsymbol{\mu}_c}|U=\boldsymbol{\mu}], \tag{5}$$

where $U = \boldsymbol{\mu}$ denotes the observed evidence that unintentional action has really occurred. With the same action content, the main difference between the original video and the counterfactual example focuses on the intention. Thus, we mitigate the negative effect brought by the training bias of action content and highlight the causal effect of intention on the model prediction by maximizing ETT of the intention.

D. Network Optimization

In this subsection, we introduce network architecture and optimization method. As shown in Fig. 8, our framework is composed of three modules including an encoder network to extract visual features, a Basic LSTM predictor to localize unintentional action, and a Siamese LSTM network to learn the causation by making a comparison between factual and counterfactual examples. The details are introduced as follows:

1) Encoder Network: Taking a video as the input, the encoder network provides the spatial-temporal features for each video frame. In practice, we can use any existing action

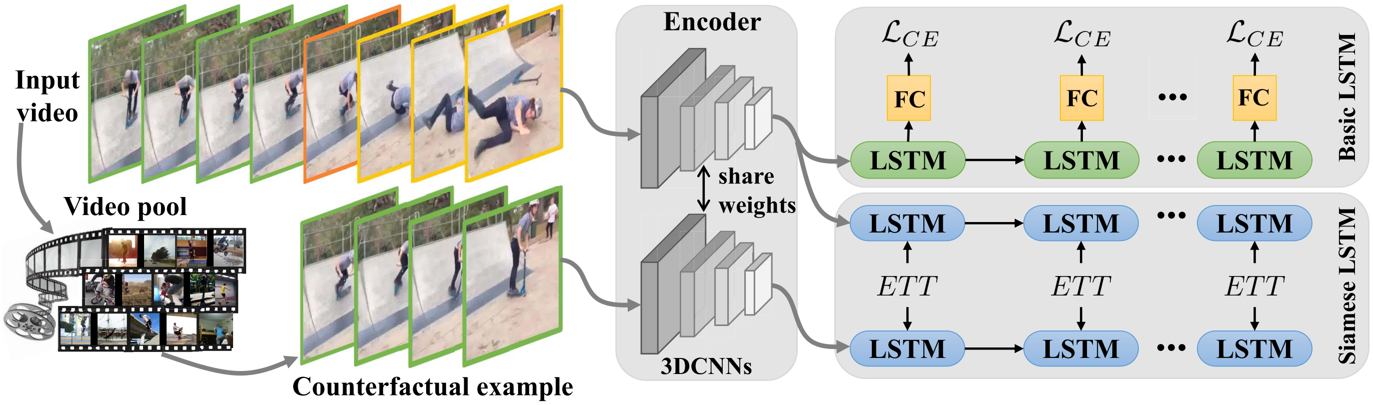


Fig. 8. The network architecture of UAL-CE. Given an unintentional action video, we generate the counterfactual example by aligning the intentional action for pairwise videos from the video pool and OOPS dataset and feed original factual video and counterfactual example into an encoder to extract spatial-temporal features for frames. Then we apply a Basic LSTM for recognizing the intentionality of action and employ a Siamese LSTM to disentangle the causal effects of both content and intention clues to learn the real causation of intention on model prediction. The weights of Basic LSTM and Siamese LSTM are shared.

recognition network as the encoder, such as the ResNet3D-18 network [70].

2) Basic LSTM: In addition to learning the spatial-temporal features of the video, we apply an LSTM as the predictor, which can be used to predict the probability of each frame. Specifically, we input a video into a many-to-many LSTM and output a sequence for hidden features, and feed the sequence features into a three-way classifier to recognize each frame as one of three categories, i.e., intentional, transitional, and unintentional.

3) Siamese LSTM: The input of Siamese LSTM is spatial-temporal features of both factual and counterfactual videos. In Siamese LSTM, we feed them into two LSTMs and make a comparison between the predictions of factual unintentional and counterfactual intentional actions to highlight the real causal effect of action intention. The weights of Basic LSTM and Siamese LSTM are shared to jointly optimize them.

During training, we optimize Basic LSTM and Siamese LSTM jointly by the following loss function:

$$\begin{aligned}\mathcal{L} &= ETT + \lambda\mathcal{L}_{CE}\\ &= ETT + \lambda\mathbb{E}_X \log P(Y|X),\end{aligned} \tag{6}$$

where L_{CE} is the cross-entropy loss used to optimize Basic LSTM while ETT is applied to optimize Siamese LSTM. λ is a hyper-parameter to balance two losses. During inference, we use Basic LSTM for prediction, since our approach only focuses on the model training.

IV. Experiments

In this section, we evaluated our approach on the OOPS dataset for two tasks, including Unintentional Action Recognition and Unintentional Action Localization. The experimental results and analysis are described in detail as follows.

A. Dataset

OOPS [18] is a recently-collected largest unintentional action dataset, which consists of 20338 videos downloaded from YouTube, adding up to over 50 hours of data. All these videos contain unintentional failures caused by various errors and environmental factors. The videos in the OOPS dataset are annotated by three workers, where each video annotation contains three time-stamps of transition to unintentional action. The OOPS dataset is split into 4674 training and 3593 testing videos since authors remove the videos without unintentional actions according to the annotation file (i.e., “transition_times.json”) to control quality. Specifically, “transition_times.json” is compressed in “Natural language descriptions.zip” and can be downloaded at the official website. “transition_times.json” is a dictionary with keys as video names and values as dictionaries, covering all videos from the training and testing sets, where each value dictionary contains the key “n_notfound” that denotes the number of workers who labeled failure “not found”. [18] removes the videos where most workers (i.e., “n_notfound≥2”) indicate there is no failure according to the annotation file. The constructed video pool contains 3004 counterfactual videos for model training, corresponding to 3004 original labeled training videos. We have intuitively displayed some examples of factual and counterfactual video pairs in Fig. 4.

B. Experiment Settings

We followed the experimental setting in [18] to evaluate the accuracy of unintentional action recognition and localization. Specifically, we applied the **same** backbone network, Kinetics pre-trained model, and supervision information for fair comparisons. Note that, although we fed the counterfactual examples for the training, no extra annotation (action state label) is introduced.

1) Unintentional Action Recognition: This task aims to recognize each video frame as one of three categories (i.e., intentional, transitional, and unintentional), which can be achieved by a three-way classification. To robustly capture spatial-temporal features of the video, we used the ResNet3D-18 [70] as the backbone network, which is pre-trained on the Kinetics action recognition dataset [71]. For the “Linear” setting in Table I, we froze the backbone network

TABLE I

COMPARISONS OF UAL-CE AND OTHER METHODS FOR THE UNINTENTIONAL ACTION RECOGNITION TASK ON OOPS

Method	Recognition Accuracy	
	Linear	Fine-tuned
PUAV-Chance [18]	33.3	33.3
PUAV-VideoSort [18]	49.8	60.2
PUAV-VideoContext [18]	50.0	60.3
PUAV-VideoSpeed [18]	53.4	61.6
PUAV-Kinetics [18]	53.6	64.0
LGF [32]	70.3	77.9
UAL-CE	**75.1**	**82.6**

TABLE II

COMPARISONS OF UAL-CE AND OTHER METHODS FOR THE UNINTENTIONAL ACTION LOCALIZATION TASK ON OOPS

Method	Localization Accuracy	
	1.0 sec	0.25 sec
PUAV-Chance [18]	25.9	6.8
PUAV-VideoSort [18]	43.3	18.3
PUAV-VideoContext [18]	52.0	25.3
PUAV-VideoSpeed [18]	65.3	36.6
PUAV-Kinetics(Linear) [18]	69.2	37.8
PUAV-Kinetics(Fine-tuned) [18]	75.9	46.7
LGF [32]	72.4	39.9
UAL-CE	**81.2**	**55.4**

and fitted our predictor whose input is the pre-trained features. For the "Fine-tuned" setting, we jointly trained the backbone network and the predictor with the annotations.

2) Unintentional Action Localization: This task is to localize the time-stamp when the action transits from intentional to unintentional. Based on the results of action intentionality recognition, we followed PUAV [18] and selected the time-stamp with the highest probability as the predicted location. According to PUAV [18], we considered the predicted location correct if the distance from any ground-truth is lower than the pre-defined threshold. As shown in Table II, we applied two thresholds provided in [18], including within 1.0 seconds and within 0.25 seconds.

C. Implementation Details

We extracted frames from each video at the same FPS (Frames Per Second, e.g., 16) and fixed the lengths of videos into the same size (e.g., 120). We applied the Kinetics pre-trained ResNet3D-18 as the backbone network to extract 512-dimension features for each video frame. We applied the same network architecture on both a Basic LSTM and a Siamese LSTM that are composed of a 256-dimension hidden layer followed by a 128-dimension fully-connected layer and share the weights. We used the Adagrad optimizer to train the model with an initial learning rate of 0.001, and set λ and batch size as 0.6 and 8, respectively. The Basic LSTM was trained by original labeled training videos of OOPS and the Siamese LSTM was trained by both original labeled training videos in OOPS and the corresponding counterfactual examples in the video pool. We used the Basic LSTM for inference. Note that the counterfactual examples are only used for training, not for testing, which has no influence on the speed of inference. We implemented UAL-CE based on the baseline of PUAV [18], where the pre-trained model is available at the official 3D-ResNets [72] implementation. We will release the code of UAL-CE, including the training and inference phases, to promote future research on unintentional action localization.

D. Results and Analysis

1) Comparison With the State-of-the-Art Methods: For both unintentional action recognition and localization tasks, we compared our approach with recent methods PUAV [18] and LGF [32]. Specifically, PUAV [18] performs a three-way classification on the video features and localizes unintentional action using a sliding window. To learn the video features, PUAV provides some self-supervised learning methods (i.e., VideoSort, VideoContext, and VideoSpeed) and a Kinetics pre-trained model (same with our approach). LGF [32] learns the goal-oriented video representations by using the extra annotations of action goals and uses the same Kinetics pre-trained model.

a) Unintentional action recognition: Table I reports the comparisons with other methods for the unintentional action recognition task on the OOPS dataset. Compared to the best performance of PUAV-Kinetics and LGF methods, our approach respectively achieved 18.6% and 4.7% improvements in the setting of "Fine-tuned", which demonstrates that our approach effectively alleviates the negative effect brought by the training bias of action content.

b) Unintentional action localization: Table II shows the comparisons with other methods for the unintentional action localization task on the OOPS dataset. It is obvious that our approach significantly outperformed PUAV and LGF methods. For example, compared with the state-of-the-art method PUAV-Kinetics (Fine-tuned), our approach achieved 5.3% and 8.7% improvements within 1.0 seconds and within 0.25 seconds, respectively. Compared with the LGF method, our approach can obtain similar performance improvements. It indicates the advantage of our approach in learning the real causation of intention on model prediction.

2) Ablation Study: To validate the effectiveness of individual components in our approach, we conducted comprehensive ablation studies with different configurations of UAL-CE for both unintentional action recognition and localization tasks on the OOPS dataset. As shown in Table III, different configurations of UAL-CE are defined as follows:

- "Baseline" indicates the baseline method that utilizes factual videos in the OOPS dataset to train Basic LSTM. The loss function is $\mathcal{L}_{CE}$ to optimize a three-way classifier supervised by $y_t|_{t=1}^{T} \in \{0, 1, 2\}$.
- "w/o alignment" indicates the method that randomly selects a part of the counterfactual video as the subsequent intentional development to generate a counterfactual example, which does not align the intentional action for pairwise videos from the video pool and OOPS dataset.
- "w/o counter+Siamese" indicates the method that feeds factual and counterfactual videos into two LSTMs without shared weights and doesn't apply the counterfactual

TABLE III

ABLATION STUDIES ON THE OOPS DATASET FOR THE TASKS OF UNINTENTIONAL ACTION RECOGNITION AND LOCALIZATION

Method	Recognition		Localization	
	Linear	Fine-tuned	1.0 sec	0.25 sec
Baseline	58.5	64.9	68.3	39.2
w/o alignment	46.7	47.1	48.6	29.5
w/o counter+Siamese	61.0	65.3	68.7	40.5
w/o counterfactual	61.5	65.8	69.9	41.7
w/o Siamese LSTM	73.2	79.5	77.6	51.7
UAL-CE	**75.1**	**82.6**	**81.2**	**55.4**

intervention, where each LSTM is followed by a fully-connected (FC) layer that trains a three-way classifier via minimizing the loss $\mathcal{L}_{CE}$.

- "w/o counterfactual" denotes the method that trains Siamese LSTM using both factual and counterfactual videos but does not conduct the counterfactual intervention. Note that "w/o counterfactual" indicates that we only use counterfactual videos as the input to train one branch of Siamese LSTM, but do not conduct the counterfactual intervention to generate counterfactual examples.
- "w/o Siamese LSTM" indicates the method that conducts the counterfactual intervention and calculates ETT to optimize the model, but two LSTMs are without shared weights during the calculation of ETT.
- "UAL-CE" is the method that conducts the counterfactual intervention to generate counterfactual examples and calculates ETT loss via Siamese LSTM.

As shown in Table III, we draw the following conclusions by comparing the experimental results:

- Compared with Baseline, UAL-CE significantly improved the performance on unintentional action recognition and localization tasks, which demonstrates the effectiveness of UAL-CE introducing the counterfactual inference to mitigate the negative effect caused by the training bias.
- The performance of "UAL-CE" and "w/o Siamese LSTM" is better than that of "w/o counterfactual" and "w/o counter+Siamese", which demonstrates that conducting the counterfactual intervention can significantly improve the performance.
- The performance of "UAL-CE" is better than that of "w/o Siamese LSTM", which demonstrates that Siamese LSTM is more effective than two LSTMs without shared weights. The same conclusion also is drawn by comparing "w/o counterfactual" and "w/o counter+Siamese".
- The performance of other variants of UAL-CE is significantly better than that of "w/o alignment", which demonstrates that aligning the intentional action for pairwise videos from the video pool and OOPS dataset can make the generated counterfactual examples more reliable and the constructed video pairs comparable.
- The improvements obtained by the counterfactual intervention are more significant than the improvements obtained by Siamese LSTM, which demonstrates the main contribution of UAL-CE is to conduct the counterfactual inference for the task of unintentional action localization.

TABLE IV

ANALYSIS ON THE NUMBER OF COUNTERFACTUAL EXAMPLES ON BOTH UNINTENTIONAL ACTION RECOGNITION AND LOCALIZATION TASKS (N: THE NUMBER OF COUNTERFACTUAL EXAMPLES)

N	Recognition		Localization	
	Linear	Fine-tuned	1.0 sec	0.25 sec
0	58.5	64.9	68.3	39.2
500	66.8	70.4	71.6	43.6
1000	69.7	76.2	75.4	45.9
2000	73.1	79.5	76.8	46.2
3004	**75.1**	**82.6**	**81.2**	**55.4**

TABLE V

STUDY OF THE HYPER-PARAMETER λ FOR UNINTENTIONAL ACTION RECOGNITION AND LOCALIZATION TASKS (REG: RECOGNITION, LOC: LOCALIZATION)

λ	Accuracy					
	0.0	0.2	0.4	0.6	0.8	1.0
Reg (Linear)	29.5	64.8	73.6	**75.1**	74.8	74.5
Reg (Fine-tuned)	32.6	68.2	76.5	**82.6**	80.3	79.9
Loc (1.0 sec)	34.7	73.5	78.3	**81.2**	80.6	80.0
Loc (0.25 sec)	21.0	43.6	51.8	**55.4**	54.9	53.8

3) Analysis on the Number of Counterfactual Examples: To investigate the effect of the number of counterfactual examples on the performance, we conducted a parameter analysis for our approach on both unintentional action recognition and localization tasks on the OOPS dataset. Table IV summarizes the performance with different parameter settings including 0, 500, 1000, 2000, and 3004 counterfactual examples. We observe significant performance improvements with increasing the number of counterfactual examples (N). For example, when N increases from 1000 to 2000, the performance on the task of unintentional action recognition achieved 3.4% and 3.3% improvements respectively in linear and fine-tuned settings. When N increases from 2000 to 3004, the recognition performance continued to grow 2.0% and 3.1% improvements respectively. We can see that increasing N can consistently improve the performance of our approach while the magnitude of the performance improvement decreases slightly with the increase of N. The above analysis demonstrates the effectiveness of counterfactual examples generated by an alignment-based method.

4) Study of the Hyper-Parameter λ: In our approach, there is a hyper-parameter λ, which makes a trade-off for L_{CE} and ETT in equation (6). We explored different λ for the tasks of unintentional action recognition and localization and presented the results in Table V. For the unintentional action recognition task, we observe that the peaks reach 75.1 and 82.6 at $\lambda = 0.6$ for the settings of "Linear" and "Fine-tuned", and tend to be flat with a slight decrease when $\lambda > 0.6$. This suggests that the Basic LSTM for the frame-wise predictions is a basic component that is important to recognize different action states (intentionality). On the contrary, the results of $\lambda = 0.0$ indicate training a dense three-way classifier contributes crucially to the model learning.

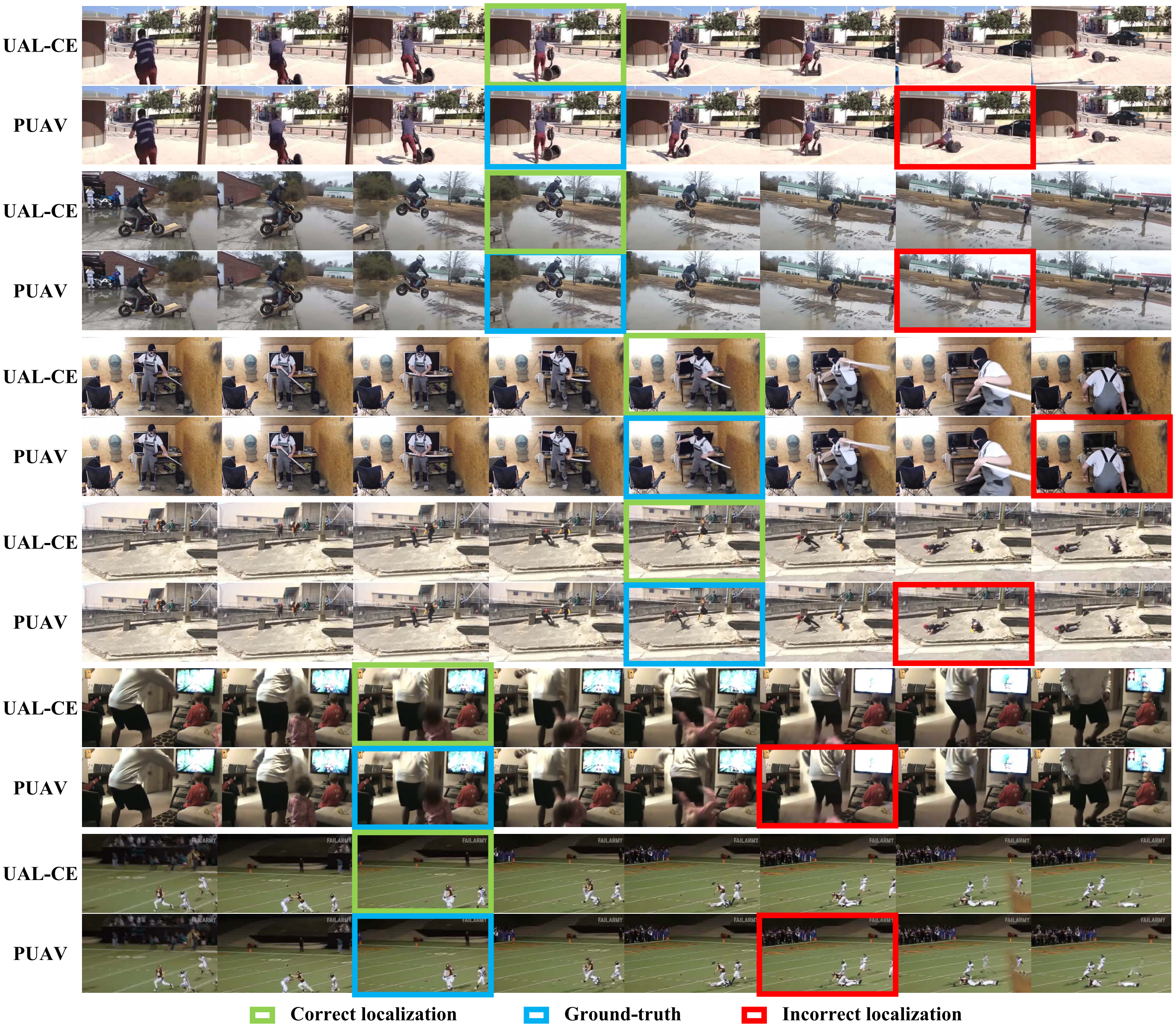


Fig. 9. The visualization comparisons of UAL-CE and PUAV on the OOPS dataset. Here, PUAV is the abbreviation of the most competitive method PUAV-Kinetics (fine-tuned). The qualitative results of several videos are shown from top to down, such as "jumping off the scooter," "playing the skateboard," "riding a motorcycle over the river," "brandishing the sword," "hitting the child," and "playing football." The green indicates the correct localization results of our approach. The blue denotes the ground truth provided by the OOPS dataset. The red is incorrect localization results of the PUAV method.

5) Visualization: To intuitively show the effectiveness of our approach, we visualized the comparisons between UAL-CE and PUAV-Kinetics. As shown in Fig. 9, for the videos like "Jumping off the scooter", "Playing the skateboard", "Riding a motorcycle over the river", "Brandishing the sword", "Hitting the child", and "Playing football", our approach correctly localized the unintentional action while the PUAV-Kinetics method fails. It demonstrates that UAL-CE can reduce the training bias and improve the accuracy of localization.

We also showed two failed examples of our approach in Fig. 10. The first example "Throwing the baby into bed" contains multiple intentionality variations, i.e., "baby being thrown into the bed" and "baby being ejected from the bed", which might lead to incorrect localization since only one intentionality variation may be recognized. For the second example "Child running under the quilt", our approach misunderstood the intentional action "running under the quilt" as an unintentional one and missed the real unintentional action "falling down". It is possible because the intentional action "running under the quilt" is hardly observed by the model.

Furthermore, we showed the attention of factual and counterfactual actions in Fig. 11, which demonstrates UAL-CE can spot the right attention and makes comparisons between them to learn the causation. Specifically, during the process of extracting spatial-temporal features of factual and counterfactual actions, we preserved the feature maps (with size 7 × 7) from the last convolution layer in 3DCNNs. The feature maps of factual and counterfactual actions are respectively denoted

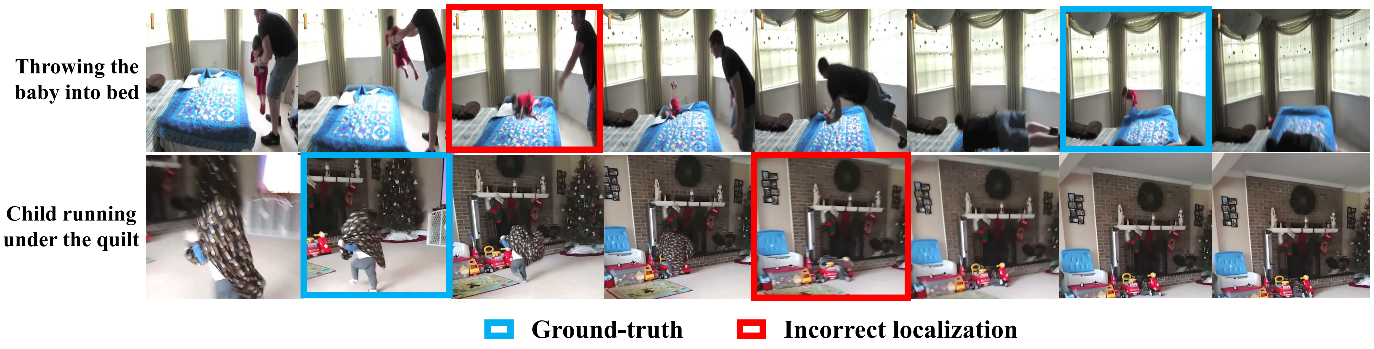


Fig. 10. Some failed examples of our approach, i.e., “throwing the baby into bed” and “child running under the quilt.”

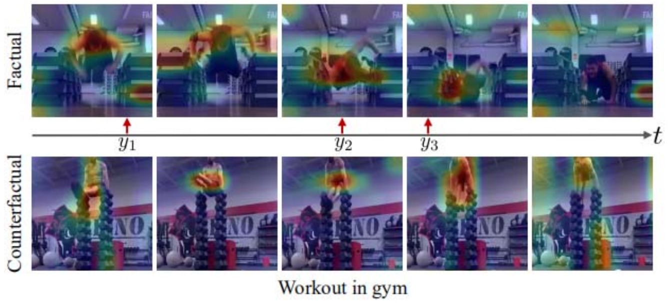


Fig. 11. The attention of factual unintentional and counterfactual intentional actions. Along the t-axis, $\{y_k\}_{k=1}^{3}$ indicate temporal annotations.

TABLE VI

PERFORMANCE COMPARISON OF UAL-CE ON THE THUMOS14 DATASET

Method	THUMOS14	
	mAP tIOU@0.5	mAP tIOU@0.7
TAL-Net [11]	42.8	20.8
G-TAD [75]	40.2	/
BSN UNet [31]	36.9	20.0
BMN [74]	32.2	/
R-C3D [30]	28.9	/
Baseline	30.8	19.7
UAL-CE	40.3	25.0

as F_μ and F_{μ_c}, both of them with the size $T \times 7 \times 7$. Then, we utilized the cross-attention model [73] to show the attention between F_μ and F_{μ_c} to focus on their semantic consistent regions. As shown in Fig. 11, the attention to factual and counterfactual actions focuses on the action itself. The highlighted factual action regions are semantic consistent with highlighted counterfactual action regions, which makes the comparisons between factual and counterfactual actions more reliable. Besides, when the action switches from one intentionality category to another, the corresponding attention is sensitive, which is beneficial to recognize intentionality variation (i.e., the transition from intentional to unintentional).

6) Discussion: We provided a discussion on the impact of UAL-CE on the general intentional action localization task via conducting experiments on the THUMOS14 dataset. Following previous efforts [11], [30], [31], [74], [75], we adopted 200 untrimmed validation videos from the validation set as the training data and utilized 213 untrimmed testing videos from the test set to evaluate the performance since only these untrimmed videos have temporal annotations. To compare with previous works [11], [30], [31], [74], [75], we followed their evaluation metrics and reported mean Average Precision (mAP) under thresholds tIoU $= \{0.5, 0.7\}$. The results of UAL-CE, baseline method, and other methods are summarized in Table VI. “Baseline” indicates the baseline method that trains Basic LSTM to learn a classifier via minimizing the binary cross-entropy loss. In UAL-CE, we first constructed counterfactual data using the OOPS dataset, and such data is obtained by matching language descriptions of unintentional actions (i.e., the annotation “goal” provided in the OOPS dataset) and intentional action categories provided in the THUMOS14 dataset. Based on intentional and unintentional action videos, UAL-CE jointly trains Siamese LSTM and Basic LSTM jointly by minimizing the loss $ETT + \lambda \mathcal{L}_{CE}$. Similar to “Baseline”, UAL-CE utilizes Basic LSTM for inference, where counterfactual data is only used for training. It can be seen that UAL-CE outperforms the baseline method and other methods under the metric mAP tIoU@0.7 and has a slight degradation compared to TAL-Net under the metric mAP tIoU@0.5. TAL-Net fuses RGB and optical flow-based features to complement each other, while UAL-CE introduces extra counterfactual data and constructs Siamese LSTM for highlighting causal effects during model learning. Therefore, TAL-Net under the metric mAP tIoU@0.5 achieves better localization performance due to introducing optical flow, while under a more strict metric (mAP tIoU@0.5) UAL-CE explores real causal effects via introducing counterfactual data during the model training and achieves better localization performance.

V. CONCLUSION

In this paper, we have proposed a causal inference approach to mitigate the negative effect brought by the training bias of action content through disentangling the causal effects between model prediction, action content, and intention clues. In our approach, we have built a video pool with intentional knowledge and conducted the counterfactual intervention to generate counterfactual examples. Then we have trained the model by maximizing the difference between the factual and counterfactual predictions to remove spurious correlations of the action content clues and highlight the intention clues. Experimental results show that our approach outperforms existing state-of-the-art methods significantly on the OOPS dataset for the tasks of unintentional action recognition and localization.

REFERENCES

[1] A. N. Meltzoff, "Understanding the intentions of others: Re-enactment of intended acts by 18-month-old children," *Develop. Psychol.*, vol. 31, no. 5, pp. 838–850, 1995.

[2] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. NIPS*, 2014, pp. 568–576.

[3] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. ICCV*, Dec. 2015, pp. 4489–4497.

[4] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. ECCV*, 2016, pp. 20–36.

[5] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proc. CVPR*, Oct. 2019, pp. 6202–6211.

[6] L. Wang, D. Q. Huynh, and P. Koniusz, "A comparative review of recent Kinect-based action recognition algorithms," *IEEE Trans. Image Process.*, vol. 29, pp. 15–28, 2020.

[7] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, and S. J. Maybank, "STA-CNN: Convolutional spatial-temporal attention learning for action recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 5783–5793, 2020.

[8] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs," in *Proc. CVPR*, Jun. 2016, pp. 1049–1058.

[9] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proc. CVPR*, Jul. 2017, pp. 5734–5743.

[10] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proc. ICCV*, Oct. 2017, pp. 2914–2923.

[11] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster R-CNN architecture for temporal action localization," in *Proc. CVPR*, Jun. 2018, pp. 1130–1139.

[12] H. Wang and L. Wang, "Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4382–4394, Sep. 2018.

[13] P. Parmar and B. T. Morris, "What and how well you performed? A multitask learning approach to action quality assessment," in *Proc. CVPR*, Jun. 2019, pp. 304–313.

[14] Y. Tang *et al.*, "Uncertainty-aware score distribution learning for action quality assessment," in *Proc. CVPR*, Jun. 2020, pp. 9839–9848.

[15] Y. Liao, A. Vakanski, and M. Xian, "A deep learning framework for assessing physical rehabilitation exercises," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 2, pp. 468–477, Feb. 2020.

[16] L.-A. Zeng *et al.*, "Hybrid dynamic-static context-aware attention network for action assessment in long videos," in *Proc. ACMM*, 2020, pp. 2526–2534.

[17] X. Yu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Group-aware contrastive regression for action quality assessment," in *Proc. ICCV*, 2021, pp. 7919–7928.

[18] D. Epstein, B. Chen, and C. Vondrick, "Oops! Predicting unintentional action in video," in *Proc. CVPR*, Jun. 2020, pp. 919–929.

[19] Y. Zhu *et al.*, "A comprehensive study of deep video action recognition," 2020, *arXiv:2012.06567*.

[20] J. Lin, C. Gan, K. Wang, and S. Han, "TSM: Temporal shift module for efficient and scalable video understanding on edge devices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2760–2774, May 2020.

[21] X. Long *et al.*, "Purely attention based local feature integration for video classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2140–2154, Apr. 2020.

[22] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "DevNet: A deep event network for multimedia event detection and evidence recounting," in *Proc. CVPR*, Jun. 2015, pp. 2568–2577.

[23] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles, "SST: Single-stream temporal action proposals," in *Proc. CVPR*, Jul. 2017, pp. 2911–2920.

[24] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," in *Proc. ICCV*, Oct. 2017, pp. 5793–5802.

[25] P. Chen, C. Gan, G. Shen, W. Huang, R. Zeng, and M. Tan, "Relation attention for temporal action localization," *IEEE Trans. Multimedia*, vol. 22, no. 10, pp. 2723–2733, Oct. 2020.

[26] R. Zeng *et al.*, "Graph convolutional networks for temporal action localization," in *Proc. ICCV*, Oct. 2019, pp. 7094–7103.

[27] L. Yang, H. Peng, D. Zhang, J. Fu, and J. Han, "Revisiting anchor mechanisms for temporal action localization," *IEEE Trans. Image Process.*, vol. 29, pp. 8535–8548, 2020.

[28] C. Lin *et al.*, "Learning salient boundary feature for anchor-free temporal action localization," in *Proc. CVPR*, Jun. 2021, pp. 3320–3329.

[29] K. Yang, P. Qiao, D. Li, S. Lv, and Y. Dou, "Exploring temporal preservation networks for precise temporal action localization," in *Proc. AAAI*, 2018, pp. 7477–7484.

[30] H. Xu, A. Das, and K. Saenko, "R-C3D: Region convolutional 3D network for temporal activity detection," in *Proc. ICCV*, Oct. 2017, pp. 5783–5792.

[31] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "BSN: Boundary sensitive network for temporal action proposal generation," in *Proc. ECCV*, 2018, pp. 3–19.

[32] D. Epstein and C. Vondrick, "Learning goals from failure," in *Proc. CVPR*, Jun. 2021, p. 11.

[33] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: A survey," *ACM Comput. Surv.*, vol. 14, p. 15, Aug. 2007.

[34] E. Jardim, L. A. Thomaz, E. A. B. da Silva, and S. L. Netto, "Domain-transformable sparse representation for anomaly detection in moving-camera videos," *IEEE Trans. Image Process.*, vol. 29, pp. 1329–1343, 2020.

[35] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. CVPR*, Jun. 2020, pp. 14372–14381.

[36] H. Lv, C. Zhou, Z. Cui, C. Xu, Y. Li, and J. Yang, "Localizing anomalies from weakly-labeled videos," *IEEE Trans. Image Process.*, vol. 30, pp. 4505–4515, 2021.

[37] R. Hinami, T. Mei, and S. Satoh, "Joint detection and recounting of abnormal events by learning deep generic knowledge," in *Proc. ICCV*, Oct. 2017, pp. 3619–3627.

[38] W. Luo, W. Liu, and S. Gao, "A revisit of sparse coding based anomaly detection in stacked RNN framework," in *Proc. ICCV*, Oct. 2017, pp. 341–349.

[39] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1992–2004, Apr. 2017.

[40] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. CVPR*, Jun. 2016, pp. 733–742.

[41] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection—A new baseline," in *Proc. CVPR*, Jun. 2018, pp. 6536–6545.

[42] J. Pearl, "Causal inference in statistics: An overview," *Statist. Surv.*, vol. 3, pp. 96–146, Oct. 2009.

[43] J. Pearl, "The seven tools of causal inference, with reflections on machine learning," *Commun. ACM*, vol. 62, no. 3, pp. 54–60, Feb. 2019.

[44] D. B. Rubin, "Essential concepts of causal inference: A remarkable history and an intriguing future," *Biostatistics Epidemiol.*, vol. 3, no. 1, pp. 140–155, Jan. 2019.

[45] J. Pearl and E. Bareinboim, "External validity: From do-calculus to transportability across populations," *Stat. Sci.*, vol. 29, no. 4, pp. 579–595, Nov. 2014.

[46] E. Bareinboim and J. Pearl, "Causal inference and the data-fusion problem," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 27, pp. 7345–7352, Jul. 2016.

[47] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proc. CVPR*, Jun. 2020, pp. 3716–3725.

[48] T. Wang, J. Huang, H. Zhang, and Q. Sun, "Visual commonsense R-CNN," in *Proc. CVPR*, Jun. 2020, pp. 10760–10770.

[49] T. VanderWeele, *Explanation in Causal Inference: Methods for Mediation and Interaction*. Oxford, U.K.: Oxford Univ. Press, 2015.

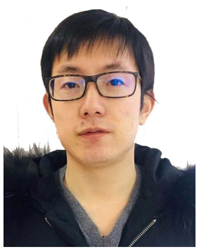
[50] J. Pearl, "Direct and indirect effects," 2013, *arXiv:1301.2300*.

[51] P. Wang and N. Vasconcelos, "SCOUT: Self-aware discriminant counterfactual explanations," in *Proc. CVPR*, Jun. 2020, pp. 8981–8990.

[52] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *Proc. ICML*, 2019, pp. 2376–2384.

[53] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *Proc. NIPS*, 2017, pp. 4066–4076.
[54] S. Chiappa, "Path-specific counterfactual fairness," in *Proc. AAAI*, 2019, pp. 7801–7808.
[55] C. Russell, M. J. Kusner, J. Loftus, and R. Silva, "When worlds collide: Integrating different counterfactual assumptions in fairness," in *Proc. NIPS*, 2017, pp. 6414–6423.
[56] K. Mohan, J. Pearl, and J. Tian, "Graphical models for inference with missing data," in *Proc. NIPS*, 2013, pp. 1277–1285.
[57] K. Mohan and J. Pearl, "Graphical models for processing missing data," 2018, *arXiv:1801.03583*.
[58] J. Vig *et al.*, "Causal mediation analysis for interpreting neural NLP: The case of gender bias," 2020, *arXiv:2004.12265*.
[59] S. Park *et al.*, "Paraphrase diversification using counterfactual debiasing," *Proc. AAAI*, vol. 33, pp. 6883–6891, Jul. 2019.
[60] S. Nair, Y. Zhu, S. Savarese, and L. Fei-Fei, "Causal induction from visual observations for goal directed tasks," 2019, *arXiv:1910.01751*.
[61] A. Forney, J. Pearl, and E. Bareinboim, "Counterfactual data-fusion for online reinforcement learners," in *Proc. ICML*, 2017, pp. 1156–1164.
[62] K. Chalupka, P. Perona, and F. Eberhardt, "Visual causal feature learning," 2014, *arXiv:1412.2309*.
[63] D. Lopez-Paz, R. Nishihara, S. Chintala, B. Scholkopf, and L. Bottou, "Discovering causal signals in images," in *Proc. CVPR*, Jul. 2017, pp. 6979–6987.
[64] E. Abbasnejad, D. Teney, A. Parvaneh, J. Shi, and A. V. D. Hengel, "Counterfactual vision and language learning," in *Proc. CVPR*, Jun. 2020, pp. 10044–10054.
[65] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen, "Counterfactual VQA: A cause-effect look at language bias," in *Proc. CVPR*, Jun. 2021, pp. 12700–12710.
[66] D. Teney, E. Abbasnedjad, and A. van den Hengel, "Learning what makes a difference from counterfactual examples and gradient supervision," in *Proc. ECCV*, 2020, pp. 580–599.
[67] J. Pearl, *Causality*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[68] J. Pearl and D. Mackenzie, *The Book of Why: The New Science of Cause and Effect*. New York, NY, USA: Basic Books, 2018.
[69] T. C. Chalmers *et al.*, "A method for assessing the quality of a randomized control trial," *Controlled Clin. Trials*, vol. 2, no. 1, pp. 31–49, May 1981.
[70] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. CVPR*, Jun. 2018, pp. 6546–6555.
[71] W. Kay *et al.*, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.
[72] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3D residual networks for action recognition," in *Proc. ICCVW*, Oct. 2017, pp. 3154–3160.
[73] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[74] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "BMN: Boundary-matching network for temporal action proposal generation," in *Proc. ICCV*, Oct. 2019, pp. 3889–3898.
[75] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-TAD: Sub-graph localization for temporal action detection," in *Proc. CVPR*, Jun. 2020, pp. 10156–10165.

Jinglin Xu received the Ph.D. degree in control science and engineering from Northwestern Polytechnical University, Xi'an, China, in 2020. Currently, she is a Postdoctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China, with a research focus on video understanding. She has broad research interests in computer vision, pattern recognition, and machine learning, where she has authored/coauthored 13 scientific papers in these areas, such as IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, CVPR, AAAI, and IJCAI.



Guangyi Chen received the B.S. and Ph.D. degrees from the Department of Automation, Tsinghua University, China, in 2016 and 2021, respectively. He has published more than ten papers on top journals and conferences, such as IEEE TRANSACTIONS ON IMAGE PROCESSING, CVPR, ICCV, and ECCV. His research interests include computer vision and machine learning, with particular expertise in attention learning, causality, and human center vision tasks such as re-identification, trajectory prediction, and action understanding.

Jiwen Lu (Senior Member, IEEE) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition. He was/is a member of the Image, Video and Multidimensional Signal Processing Technical Committee, the Multimedia Signal Processing Technical Committee, and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society; and a member of the Multimedia Systems and Applications Technical Committee and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society. He is a fellow of IAPR. He was a recipient of the National Natural Science Funds for Distinguished Young Scholar. He serves as the General Co-Chair for the International Conference on Multimedia and Expo (ICME) 2022 and the Program Co-Chair for the International Conference on Multimedia and Expo 2020, the International Conference on Automatic Face and Gesture Recognition (FG) 2023, and the International Conference on Visual Communication and Image Processing (VCIP) 2022. He serves as the Co-Editor-of-Chief for *Pattern Recognition Letters* and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE, and *Pattern Recognition*.

Jie Zhou (Senior Member, IEEE) received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From 1995 to 1997, he served as a Postdoctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China, where he has been a Full Professor with the Department of Automation since 2003. In recent years, he has authored more than 100 papers in peer-reviewed journals and conferences. Among them, more than 30 papers have been published in top journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and CVPR. His research interests include computer vision, pattern recognition, and image processing. He is a fellow of IAPR. He received the National Outstanding Youth Foundation of China Award. He is an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and two other journals.