ProTAL: A Drag-and-Link Video Programming Framework for Temporal Action Localization

Yuchen He
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, Zhejiang, China
heyuchen@zju.edu.cn

Jianbing Lv
School of Software Technology, Zhejiang University
Hangzhou, Zhejiang, China
lvjianbing@zju.edu.cn

Liqi Cheng
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, Zhejiang, China
lycheecheng@zju.edu.cn

Lingyu Meng
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, Zhejiang, China
kevinmeng@zju.edu.cn

Dazhen Deng*
School of Software Technology, Zhejiang University
Hangzhou, Zhejiang, China
dengdazhen@zju.edu.cn

Yingcai Wu
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, Zhejiang, China
ycwu@zju.edu.cn

Abstract

Temporal Action Localization (TAL) aims to detect the start and end timestamps of actions in a video. However, the training of TAL models requires a substantial amount of manually annotated data. Data programming is an efficient method to create training labels with a series of human-defined labeling functions. However, its application in TAL faces difficulties of defining complex actions in the context of temporal video frames. In this paper, we propose ProTAL, a drag-and-link video programming framework for TAL. ProTAL enables users to define **key events** by dragging nodes representing body parts and objects and linking them to constrain the relations (direction, distance, etc.). These definitions are used to generate action labels for large-scale unlabelled videos. A semi-supervised method is then employed to train TAL models with such labels. We demonstrate the effectiveness of ProTAL through a usage scenario and a user study, providing insights into designing video programming framework.

CCS Concepts

• **Human-centered computing** → *Interaction design*; **Systems and tools for interaction design**.

Keywords

Interactive Data Programming, Data Annotation, Temporal Action Localization

ACM Reference Format:
Yuchen He, Jianbing Lv, Liqi Cheng, Lingyu Meng, Dazhen Deng, and Yingcai Wu. 2025. ProTAL: A Drag-and-Link Video Programming Framework for Temporal Action Localization. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26-May 1, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3706598.3713741

*Dazhen Deng is the corresponding author.

CHI '25, April 26-May 1, 2025, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713741

1 Introduction

Temporal Action Localization (TAL) is an important task within the field of computer vision, particularly for understanding and indexing long videos [5, 12, 71, 74, 77, 78]. TAL aims to detect the start and end timestamps of specific actions and their categories [60]. In real-world scenarios, most videos are untrimmed, and the actions of interest may only appear in a small portion of frames. Therefore, compared to video-level action classification [4, 62], TAL faces the challenge of temporally localizing actions while ignoring irrelevant frames and distracting backgrounds.

With the rapid development of computer vision techniques, deep learning-based methods [33, 52, 74] have achieved commendable results on various TAL benchmarks, such as ActivityNet-1.3 [14] and THUMOS14 [25]. However, training deep neural networks for TAL often requires a large amount of annotation data on specific videos, the acquisition of which incurs significant labor costs. While single-frame supervision [37, 69] and semi-supervision [39] settings have been introduced to train TAL models with fewer annotations, these methods still involve a tedious annotation process, annotators are required to label each sample individually and cross-validate the results, which remains time-consuming and labor-intensive.

As a key approach in data-centric AI [73], data programming [46, 48] injects human knowledge into data to generate labels for model training. Although these labels can be noisy, they are crucial for the initial training of deep learning models. Data programming typically involves two stages: decomposition and reconstruction. During decomposition, experts use pretrained models to generate initial labels. In reconstruction, they define labeling functions to create new labels based on these initial ones. For example, in image semantic segmentation, experts might use a pretrained model to identify segments like "transportation" and "water" in a new dataset. They can then define relations between segments, such as transportation above water, to extract the label "boat" for model training [22].

Despite its success in natural language and image processing, applying data programming to TAL presents significant challenges.

First, decomposing actions into meaningful substructures is difficult because actions in videos are spatiotemporal data, adding complexity. For example, baseball throwing involves finer actions like hip turning and hand movement, but pretrained models often lack the accuracy to localize these atomic actions. **Second**, the spatiotemporal nature of actions makes it challenging to define labeling functions that capture detailed action dynamics. Human actions involve complex relationships between human poses and objects across frames, requiring an effective method to translate conceptual actions from users' minds into accurate labels.

To address the first challenge, we propose ProTAL, a TAL data programming framework with multiple levels of action decomposition. The framework first breaks down actions into key events, which are then defined by fine-grained visual elements extracted by computer vision modules that recognize human poses and objects frame by frame. The design is inspired by the observation that humans can identify ongoing actions from just a few frames, thanks to discriminative cues within the action, which we refer to as key events.

To address the second challenge, we propose a drag-and-link interaction design that enables users to define key events efficiently using a graph-based visualization. Human poses and objects are mapped to nodes on a canvas, where users can drag, link, and constrain angles between key nodes to specify relations and define key events. The design also supports smooth visual transitions from real video frames to key nodes, allowing for intuitive abstraction and definition of key events.

We developed a system to implement the proposed framework and drag-and-link interaction. After uploading a video dataset, the computer vision modules will extract human poses and objects automatically. Then, users can select the videos of interest to define key events. The human poses and objects of each frame are represented as nodes and links, which are interactive and editable. Users can select specific frames as key events and complete the definition with drag-and-link interaction. The key events defined are applied to the rest of the videos, generating frame-wise action labels for the dataset. The labels are used to train TAL models, and the models are then applied to the dataset, which accelerates the whole process of data annotation. ProTAL also visualizes the distribution of key events across the dataset and helps further fine-tune the annotation. With several iterations, ProTAL helps users create an initial dataset for model training. The effectiveness of our framework and interaction design was demonstrated in a practical usage scenario and a user study. The main contributions of this paper are as follows:

- We propose ProTAL, a video programming framework that decomposes complex human actions into key events and atomic elements for flexible data programming.
- We design an intuitive drag-and-link interaction that quickly translates user concepts into data programming rules.
- We implement a system of ProTAL that facilitates TAL annotation and training, demonstrating the effectiveness of our framework and interaction design.
- We gain insights into interactive video programming and offer lessons for designing TAL annotation systems through controlled user studies with ProTAL.

2 Related Work

We review previous works on TAL, interactive annotation of video data, and data programming.

2.1 Temporal Action Localization

Under the wave of the deep learning era, the field of TAL has undergone revolutionary development. Leveraging the robust video backbones such as C3D [58], I3D [4], and VideoMAE [57], the technology for TAL has made significant strides. Currently, TAL primarily operates under two settings: full supervision and weak supervision.

Fully-supervised TAL is the most fundamental setting, utilizing the most labeled information for model training. The earliest work can be traced back to the detection of actions by classifying sliding window proposals [54]. Subsequently, the anchor mechanism was introduced to enhance the flexibility of proposal regions [17]. With the introduction of TAL-Net [5], the workflow of TAL was further refined, evolving the anchor mechanism into a two-stage approach. Similarly, ActionFormer [74] and TriDet [53] have enhanced TAL performance. For weakly-supervised TAL, UntrimmedNet [61] is an pioneering work, consisting of a classification module and a selection module to infer the temporal boundaries of action instances. STPN [40] introduced sparse regularization for video-level classification. Nguyen et al. [41] and Liu et al. [34] made effective use of background segments to enhance the accuracy. Other settings like single-frame supervision [29, 37, 69] have been proposed to reduce annotation costs. This setting lies between fully supervised and weakly supervised, as start and end timestamps are not required for training. Instead, the model can be trained with just one annotated frame per action segment [37] or background segment [69].

Regardless of the type of supervision, state-of-the-art TAL methods have achieved impressive performance across various benchmarks. However, a significant gap persists between these methods and practical applications. These models often face the problem of "data hunger". Training a TAL model typically requires a large-scale annotated dataset, and obtaining these annotations requires considerable costs. While weakly supervised and single-frame supervised methods can partially mitigate this challenge, the annotation process still requires manually reviewing each video, making it time-consuming and ultimately not scalable.

2.2 Interactive Annotation of Video Data

With the increasing demand for automatic video analysis and understanding in industries such as manufacturing, education, and sports, the high cost of video annotation has become a key barrier to applying these models. To address this challenge, researchers in the fields of human-computer interaction have proposed various interactive video annotation frameworks. Using rules or machine learning algorithms, these frameworks significantly reduce workload, offering an effective solution.

Kurzhals et al. [28] utilized video segmentation algorithms to divide eye-tracking data into multiple segments and then cluster them, enabling users to annotate multiple segments simultaneously. HistoryTracker [42] employed historical data and algorithms to hot-start the annotation system, allowing baseball tracking data to be generated with minimal user input. According to the needs of racket

sports analysts, EventAnchor [10] proposed a multi-level video annotation framework that integrates computer vision algorithms and extensive domain knowledge, facilitating efficient exploration of video content. VideoModerator [56] is a system developed to annotate anomalous videos, which first recommends videos through a classifier and then provides users with three different views to analyze and annotate these recommendations. ActLocalizer [6], tailored for TAL tasks, helps users expand single-frame annotations to full supervision by aligning action instances with a storyline-based view, thus improving the accuracy of TAL.

However, despite the significant improvements these frameworks have made in enhancing annotation efficiency, they still face challenges when applied to TAL. Firstly, although these frameworks offer well-designed user interfaces to help users understand and explore data, they are often tailored to specific tasks or scenarios. Moreover, even with these frameworks, each video still requires handling for annotation or validation, limiting scalability. It means that constructing large-scale datasets still requires substantial time and labor. Secondly, while ActLocalizer [6] presents a method that allows users to enhance supervision in datasets with single-frame annotations, it is still not suitable for scenarios where the dataset needs to be built from scratch.

2.3 Data Programming

Data programming offers a scalable paradigm that allows users to quickly build large datasets from scratch for model training. As one of the most promising approaches within data-centric AI, data programming injects knowledge into data in the form of user-defined labeling functions, enabling the generation of annotated data more efficiently than manually labeling each sample individually. Data programming was first explored in the field of natural language processing [2, 47, 59]. Snorkel [46] enables users to provide higher-level supervision in the form of labeling functions. This approach allows for the creation of large-scale datasets without the need to meticulously manage the resulting noise and conflicts. Ruler [13] and TagRuler [8] enable users to efficiently obtain accurate labeled data to generate labeling functions using predefined concepts and highlighting keywords, simplifying the design of labeling functions.

Researchers have been working to expand the application scenarios of data programming. However, there are still relatively few applications in computer vision. Visual Concept Programming [22] was the first to extend data programming to image data. This approach begins by training a self-supervised model to extract visual concepts and then offers an interactive interface that allows users to create labeling functions without writing code, enabling iterative model training. It lacks the ability to define dynamic concepts, making it unsuitable for video data. Additionally, VideoPro [21] applies data programming to video data through sequence pattern mining, but fails to provide temporal annotations for actions, limiting its utility in TAL. To address these limitations, we propose a novel framework that extends data programming to TAL, aiming to bridge the gap between TAL methods and practical applications.

3 Problem Formulation

We first introduce the concepts of data programming and how we formulate the problem of data programming in the TAL scenario.

Data Programming Paradigm. To begin, we introduce the paradigm of data programming, which usually consists of two stages. The first stage involves the automatic extraction of visual elements. Advanced computer vision algorithms are used to extract visual elements that may serve as candidates for the definition of new labels. The second stage focuses on defining the rules that can be used to compose the candidates together and generate new labels.

The key to effective data programming in TAL is to extract basic action elements and reconstruct them. In this study, we first decompose actions into key events inspired by the concept of "key frames" in video editing, which define the start and end points of transitions or animations. While key points can anchor human actions, we use the term "key events" instead of "key frames" because a key event can span several frames. This flexibility accounts for slight variations in the same action across different videos, where a single key frame would be too restrictive. Key events are considered the bridge between the target actions and basic visual elements.

Key Event. A key event is an atomic event within an action characterized by changes in the relations between several visual elements, which is easier to decompose and define. For example, the "clean and jerk" action includes a key event K_0 (Figure 2E): "The barbell moves from below the athlete's head (Figure 2E1) to above the athlete's head (Figure 2E2)."

Key events serve as anchor points for the actions, but another unresolved problem is how to define and refine the key events using low-level visual elements. Taking the case in Figure 2 as an example, K_0 involves two visual elements: the "barbell" and the "person's head," with K_0 being defined by the relative position change between these two visual elements. However, to leverage these visual elements to define key events, two key questions remain to be addressed:

- **Q1** What **visual elements** should be extracted for the definition of key events?
- **Q2** What **constraints** are required to define a key event with visual elements?

4 Design Considerations of ProTAL

To prepare for the design of ProTAL, we conducted a literature review and a workshop study[1] to identify the space for visual elements and constraints.

4.1 Literature Review

Since a key event is a temporal and spatial substructure of an action, understanding the visual elements involved in key events requires first identifying the visual elements associated with actions. To explore this, we conducted a literature review to gain insights from previous research on human action recognition and detection. We reviewed 23 studies [1, 3, 9, 11, 15, 16, 18, 19, 26, 27, 31, 43–45, 55, 63, 65, 68, 70, 72, 76, 79, 81] and identified two main categories of action-related visual elements. Based on the interactions involved in the actions, actions can be categorized into three categories: (1) **single-human actions**, (2) **human-human interaction**, and (3) **human-object interaction**, and focusing on these

[1]The study has been approved by State Key Lab of CAD&CG, Zhejiang University.

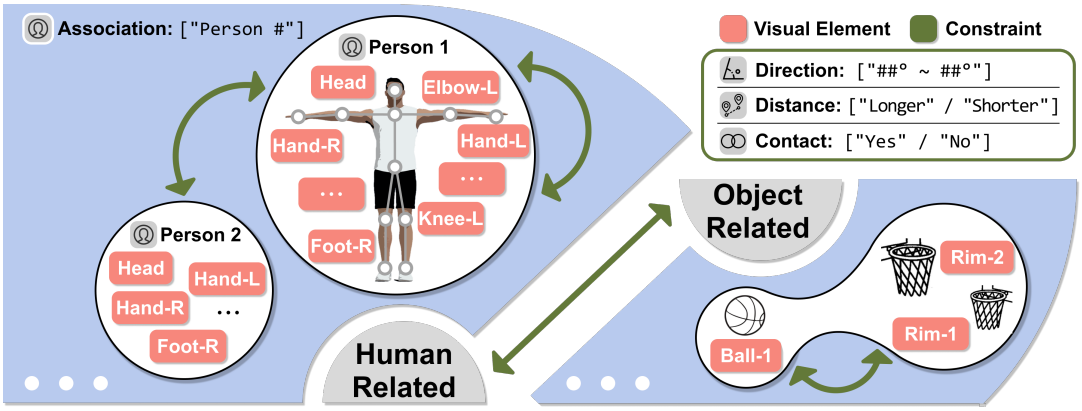

Figure 1: The space of visual elements and constraints in key event definitions. Visual elements include two categories: human-related visual elements, mainly human body parts, usually represented as skeletons; and object-related visual elements, including objects involved in the action. Constraints include direction, relative distance, contact, and association constraint.

categories, many studies have tried to improve the action detection or recognition performance. It is worth noting that object-object interactions could be considered a separate category, but they are beyond the scope of this discussion. In practice, objects can be highly complex. For instance, a modern car can be broken down into components like pistons, crankshafts, and valves, each operating with a distinct mechanism of motion. Providing a formal definition that encompasses all types of objects is inherently challenging. Additionally, a human pose can be viewed as a simplified representation of a machine. If the target object is clearly defined, the proposed method for modeling human-human interactions could be adapted and extended to handle such scenarios. Therefore, we focus on the discussion on single-human actions, human-human interaction, and human-object interaction. In these categories, the interaction subjects considered are actually humans and objects. Therefore, we can start from these two interaction subjects and consider the visual elements related to the action: human-related visual elements and object-related visual elements.

Human-related Visual Elements. Human-related visual elements are central to actions, as the human body plays a leading role in action involving multiple body parts. Among the 23 studies, 19 utilize pre-recognized human bodies as input, with 10 in the form of poses and 9 in the form of bounding boxes. Therefore, when considering human-related visual elements, it is essential to account for the various parts of the human body.

Object-related Visual Elements. In the context of human-object interaction, out of 21 studies that addressed this area, 11 utilized bounding boxes of relevant objects as input, in addition to learning representations directly from RGB. Thus, for object-related visual elements, we need to focus on objects that are relevant to the action being performed.

4.2 Workshop Study

Through our literature review, we identified the potential types of visual elements involved in key events and answered Q1. The next step is to determine the types of relations between them should serve as constraints in key event definitions. As the concept of a key event is newly introduced in this paper, it may not be appropriate to apply element relations considered in existing action-related works.

A key event is characterized by changes in the relations between several visual elements, which can be represented as a series of **state** transitions. As illustrated in Figure 2, states 1 and state 2 correspond to two distinct states within the key event K_0, allowing K_0 to be expressed as $K_0 \coloneqq state_1 \longrightarrow state_2$. It is apparent that each state, such as states 1 and 2, can be represented by a frame in the action, indicating that a key event is, in fact, a dynamic concept composed of a sequence of static states. Therefore, when defining a key event, we are essentially defining a series of static states. Therefore, the relations between the visual elements in these states are also static. This strategic decomposition of key events significantly simplifies their retrieval, as it only requires identifying static frames that match the specified rules.

The nature of key events guides us in further exploring the constraint space. Following this, we conducted a workshop study with a brainstorming session and a follow-up seminar to derive the space of the constraint in detail.

Participants. We conducted the workshop with 8 action annotators (E1-E8) who have participated in action annotation more than 5 times and have backgrounds in programming and AI. Among them, E2 and E7 (both male) are Ph.D. in computer science, while the others are graduate students (4 in computer science and 2 in sports science, male=4, female=2). All participants have experience in action annotation for racket sports (e.g., tennis, table tennis, badminton), 75% have experience with other ball sports (e.g., basketball, football, volleyball), and 50% have annotation experience with other types of actions.

Procedure. We began by assessing the participants' backgrounds and understanding the types of action they had previously annotated. Next, we introduced the concept of key events and the visual element space derived from our earlier research. After ensuring that the participants had a good understanding of the relevant concepts, we organized a brainstorming session in which each participant was shown three videos: one containing jumping jacks (single human

action), one containing handshake (human-human interaction), and another containing clean and jerk (human-object interaction). Each video contains more than 10 action instances. These actions involve multiple types of relations, including those between human-related elements, object-related elements, and between human-related and object-related elements, effectively covering all possible pairings of element types. These actions are also common to minimize potential bias due to varying levels of familiarity and to facilitate broader discussions. Participants were asked to propose a key event for each action and, assuming they had access to the bounding boxes of all action-related visual elements in the frames, provide a pseudocode (or a natural language description) that could be used to retrieve the frames corresponding to the key event, each video for 20 minutes. Following this, we held a seminar where participants summarized the 24 pieces of pseudocode and identified the type of constraints needed to define key events.

Findings and Discussions. All participants highlighted the importance of relative position between visual elements. Given that each frame naturally provides the bounding box position of visual elements, relative position becomes a key consideration when defining relations between them. Also, relative position is a very intuitive relation for a pair of visual elements. To express the relative position, such as "above," "to the left," "upper right," "upper left," etc., 83% of the pseudocode examples calculated the direction angle, while 58% involved directly comparing x-coordinates or y-coordinates. During the seminar, it was agreed that while direct coordinate comparison might be feasible for simpler direction relations such as "above" and "below", calculating the direction angle offers broader coverage and greater accuracy.

In addition to direction, participants also mentioned distance as a crucial aspect of relative position. E1 and E4 noted that using absolute pixel distance is impractical, as variations in camera shooting distance and changes in viewing angle can cause this value to fluctuate, so they opted for relative distance, comparing the magnitude of the distances between pairs of visual elements. Participants noted that direction and relative distance together were sufficient to describe a relative position. Furthermore, these relations can be applied between any type of visual element.

Beyond relative position, it was observed that in the pseudocode for the second action, all participants utilized the intersection of the bounding boxes of two individuals' hands. Participants agreed that contact is a required constraint, and the overlapping of regions can capture this relation better than distance because objects vary in size and shape. Furthermore, E2 proposed that the association constraint, which defines the relationship between body parts and their respective individuals, is essential. This association can be derived from the extracted human poses. All participants agreed that in scenarios involving multiple individuals, accurately associating body parts with the correct individuals is critical for identifying and defining key events.

4.3 Design Principle

Our goal is to design a TAL data programming framework that allows users to define key events through interaction and use these rule-based definitions to generate labels for unlabeled video sets to train the TAL model. Based on the previous research, we now have a clear understanding of the space of visual elements and constraints, shown as Figure 1.

Visual elements. There are two categories of visual elements to consider: human-related visual elements and object-related visual elements, where the human-related visual elements involve various body parts. Therefore, when implementing the system, it is essential to provide:

- **P1** Automatic extraction of visual elements in frames, including human body parts and action-related objects.
- **P2** Supporting direct manipulation [23] of the visual elements on the user interface.
- **P3** Providing intuitive visual mapping of visual elements from video frames to canvas.

Constraints. For constraints, it is necessary to provide the relative position relations, including direction (angle) and relative distance. In addition, contact relations, which indicate whether two visual elements are in contact, and the association constraint, which constrain the person to whom a human-related visual element belongs, should also be provided. Therefore, the design principles for constraints include:

- **P4** Providing sufficient constraint candidates, including direction, relative distance, contact, and association constraint.
- **P5** Supporting interactive setting of constraints to define key events, with visualization of constraints on the user interface.
- **P6** Enabling users to get feedback on the generated labels and iteratively fine-tune the constraints they set.

5 Framework of ProTAL

We propose ProTAL, a data programming framework designed for TAL. Built on the data programming paradigm, ProTAL incorporates the unique characteristics of temporal action data in TAL. The framework allows users to efficiently generate training labels for unlabeled videos through interaction. As shown in Figure 2, ProTAL follows a three-stage pipeline, which is described in detail below.

5.1 Extraction of Action-Related Visual Element

For unlabeled video data, the first stage of ProTAL involves extracting action-related visual elements from each frame. These visual elements are then used to filter frames that contain a specific set of elements that meet defined constraints. According to P1, visual elements extracted are categorized into two groups: human-related elements and object-related elements. Using advanced computer vision models, both categories can be extracted automatically and efficiently.

For human-related elements, it is necessary to extract various body parts of the human and to distinguish which person these elements belong to (P4). Existing human pose estimation methods, such as ViTPose [66] and RTMPose [24], can be employed to obtain skeleton information from each frame, thereby capturing the location of different body parts for each person in the frame. For object-related elements, state-of-the-art object detection and semantic segmentation models are highly effective in detecting or segmenting specific objects in videos, thus providing the necessary location information (P4). In addition to these models, recent advances in multimodal models, such as Grounding DINO [32] and

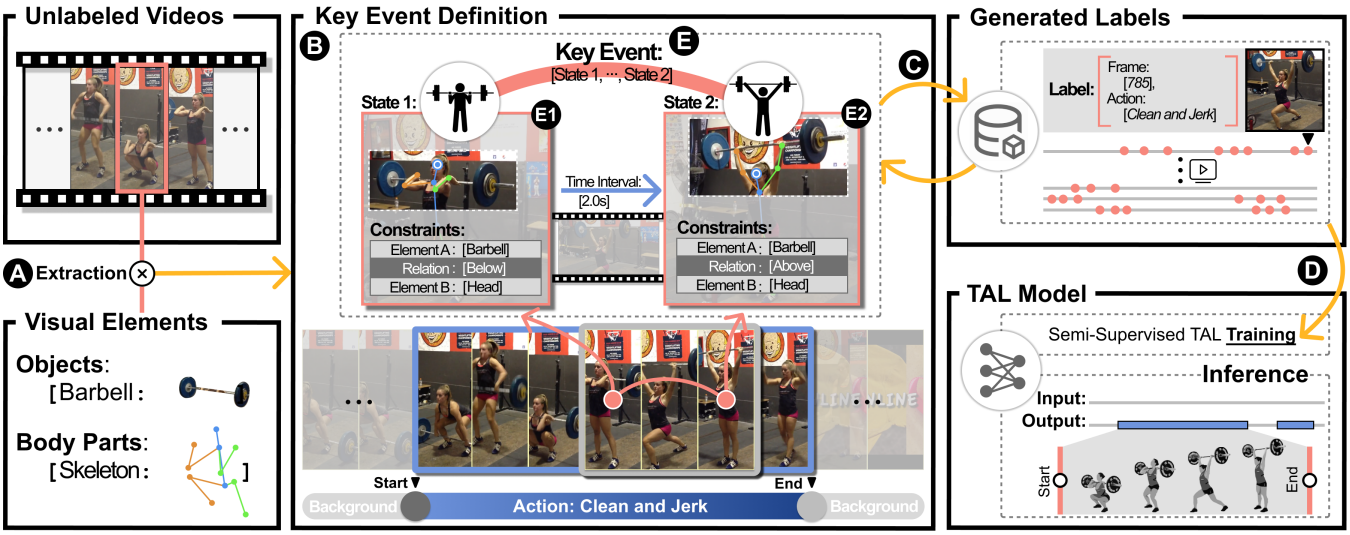

Figure 2: Framework of ProTAL. The first stage is (A) the automatic extraction of action-relevant visual elements. The second stage is (B) the defining of key events based on interactions, followed by (C) the generation of key event labels. The third stage is (D) the model training with a semi-supervised TAL method based on the generated labels.

Grounded SAM [49], combine the strengths of various types of models to enable more robust detection and segmentation of complex visual elements using natural language prompts. These models are also useful for extracting object-related elements from videos. ProTAL's design allows flexibly integrating computer vision models that best suit users' needs. For instance, users can integrate a detection model that is purposely trained for tennis balls to extract their positions more precisely compared to using general vision large models, such as Grounding DINO.

At this stage, ProTAL has extracted action-related visual elements from the initial unlabeled video set. For any given frame f in any video v within the video set, the visual elements extracted from f are denoted as ELM_f:

$$ELM_f = \{e_1, e_2, \cdots e_n\}, \tag{1}$$

$$e_i \coloneqq \{Type, Position, Association(\text{human-related}), \cdots\}, i \in \{1, \cdots, n\}, \tag{2}$$

where each e represents a visual element, which includes attributes such as location and category.

5.2 Key Event Definition and Label Generation

After the automatic extraction of visual elements, the second stage involves defining key events through an interactive interface. These key event definitions serve as rules for identifying frames that correspond to the key events and assigning labels to them. The labels are then presented to the users, enabling them to refine the key event definitions to improve label quality.

5.2.1 The Concept of Key Event. In order to define a key event, denoted by K, users are required to specify n_s, the number of states that comprise K:

$$K \coloneqq state_1 \rightarrow state_2 \rightarrow \cdots \rightarrow state_{n_s}, \tag{3}$$

and the threshold thr of time interval between adjacent states:

$$state_k \xrightarrow{t \leq thr_{k,k+1}} state_{k+1}. \tag{4}$$

For each state, users are required to provide a detailed definition. In order to define $state_k$, users are required to specify the visual elements involved, the attributes of each of these elements, and the relations between them:

$$state_k \coloneqq \{ELM, REL\}, \tag{5}$$

$$ELM = \{e'_1, e'_2, \cdots, e'_{n_e}\}, \tag{6}$$

$$e'_i \coloneqq \{Type, Association(\text{human-related}), \cdots\}, i \in \{1, \cdots, n_e\}, \tag{7}$$

$$REL = \{r'_{i,j}, \cdots\}, i, j \in \{1, \cdots, n_e\}, \tag{8}$$

$$r'_{i,j} \coloneqq \{Value_1, Value_2, \cdots\}, i, j \in \{1, \cdots, n_e\}, \tag{9}$$

where e'_i denotes the element i involved in the state definition, n_e denotes the number of such elements, and $r'_{i,j}$ denotes the user defined relation between element i and element j. The set of values $\{Value_1, Value_2, \dots\}$ corresponds to the specific parameters or attributes for the corresponding type of relation.

5.2.2 The Retrieval of Key Event Frames. When users complete the definition of a key event, the frames in the videos that match the user-defined key event definition will be retrieved and assigned labels. Specifically, in each state within the key event, the visual elements and the constraints together serve as the rules for searching through each frame in the video to identify those that align with the state's definition. After retrieving the frames corresponding to each state, the sequence of frames that meet the conditions based on the user-defined time interval threshold thr between the states represents the frames of the key event. Thus, retrieving key event frames in the video primarily involves retrieving frames that satisfy

the definitions of each state within the key event. First, we represent frames in the video abstractly. Given all visual elements ELM_f extracted from a frame f and all computable relations between them $REL_f = \{r_{i,j}, \cdots\}$, f can be structurally represented as a graph, denoted as $G_f \coloneqq \{ELM_f, REL_f\}$, since each visual element can be treated as a node with attributes and each relation between a pair of nodes can be considered as an edge with weights. This structure aligns with the state definition $G_{state_k} \coloneqq \{ELM, REL\}$.

Given the state definition $state_k$, as ELM_f may contain redundant visual elements, determining whether f is a frame corresponding to $state_k$ requires checking if G_{state_k} is a subgraph of G_f. This means that determining whether a frame satisfies the state definition is essentially a subgraph matching problem with edge weights.

Since state definitions are generally not overly complex and the number of nodes in the subgraph is typically small, a search algorithm with pruning, denoted as Φ, can be employed for subgraph querying:

$$\Phi(G_f, G_{state_k}) = \begin{cases} True & f \text{ corresponds to } state_k, \\ False & \text{otherwise.} \end{cases} \tag{10}$$

When $\Phi(G_f, G_{state_k})$ = True, the frame f corresponds to $state_k$; otherwise, it does not. After labeling all frames corresponding to each key event, the results are presented to the users, guiding them to refine the key event definitions in order to generate more accurate labels for TAL training.

5.3 TAL Model Training

After completing the first two stages, the original video dataset now contains sparse frame-wise action labels. The objective of this stage is to utilize these frame labels to train the TAL model.

5.3.1 Problem Statement. Given a video v with T frames, with an action instance in v from $[t_l : t_r]$, where $0 \le t_l \le t_r \le T$. Since the key event is a substructure of the action, the frames labeled by states of a key event lie within the action. The generated labels for the action instance consist of several frames between t_l and t_r, denoted as $Label_{ProTAL} = \{t_1, t_2, \cdots, t_m\} \subseteq \{t_l, \cdots, t_r\}$, with each labeled frame implicitly assigned an additional state label. This differs from full supervision labels, $Label_{full} = \{t_l, \cdots, t_r\}$, which include all frames within the action instance, and from the single-frame supervision labels used in SF-Net, $Label_{SF} = \{t'\}, t' \in \{t_l, \cdots, t_r\}$, where only one frame within the action instance is labeled. Furthermore, in both full supervision and single-frame supervision, every action instance is assigned labels. For ProTAL, however, there may be instances that remain unlabeled.

5.3.2 Training Method. ProTAL employs a semi-supervised approach by extending SF-Net to train with $Label_{ProTAL}$. SF-Net can be trained with any number of frame labels, but cannot fully leverage unlabeled samples for representation learning. To address this, we refine the classification target of the classification head to the state level. Given that the states within key events are inherently ordered, a state order loss on unlabeled videos is introduced during training to penalize any incorrect prediction of state order.

6 Interface Walkthrough: A Practical Scenario

Based on the proposed ProTAL framework in section 5, a prototype system with a drag-and-link interactive user interface was implemented, as shown in Figure 3. In this section, we present a practical usage scenario where the system is used to program an unlabeled table tennis video dataset for TAL training. We demonstrate how the user interact with the system throughout the process and evaluate the final TAL model.

Background. *Alex* is a data analyst with extensive experience in annotating table tennis action data. He has participated in the annotation of a number of table tennis-related datasets and is proficient in the use of AI methods to identify objects such as balls, players, tables, and actions in video. The current method for segmenting rallies in table tennis match videos relies on identifying score changes on the scoreboard. However, this approach is sometimes inaccurate due to the delays in score adjustment during broadcast. To address this issue, Alex aims to train a TAL model that can temporally locate table tennis serve actions, with the objective of refining rally segmentation by detecting the time intervals of serve actions.

Implementation Details. The prototype system uses several computer vision modules to extract visual elements. For human-related elements, RTMPose [24] is integrated to extract human poses from videos. For object-related elements, such as the ball and the table, an off-the-shelf detection model trained specifically for table tennis analysis tasks is utilized. According to Equation 2, the extracted attributes of visual elements include position, type, and association (derived from human poses). To track individuals and objects across different states of a key event instance, we use the Intersection over Union (IoU) of bounding boxes of adjacent frames, given the short time span. This ensures that when matching subgraphs, individuals with the same ID in each G_{state} correspond to the same person in the video. For relative distance, during subgraph matching, we ensure that the length order of each corresponding edge pair remains consistent with the definition. For the contact constraint, two bounding boxes are considered to be in contact if their IoU exceeds a predefined threshold.

6.1 User Interface Overview

Functionality. The user interface includes five views. *Dataset View* supports video browsing and label review. *Event View* allows key event management. *Defining View* displays a canvas for defining key events. *Frame View* lists the frames retrieved based on the user-defined key events. *Training View* displays the status of model training.

Interaction. The drag-and-link interaction design is inspired by motion editing techniques in animation. In animation editing, keyframes are often manipulated by dragging human joint points to create or adjust motion sequences, as demonstrated in systems like TimeTunnel [80] and the pin-and-drag interface [67]. Additionally, ProTAL abstracts each state within a key event as a graph, making drag-and-link interactions a natural fit for defining states. Dragging provides an intuitive way to adjust nodes [64] or subgraphs [50] within the graph, while link is an inherent component of the graph [20, 30, 38], effectively representing the relations between nodes. This design ensures that defining relations between nodes through linking is intuitive.

Figure 3: System screenshot. Users can navigate the video dataset and identify key events in *Dataset View* (A). They can add key events in *Event View* (B) and define them through drag-and-link interactions in *Defining View* (C). The distribution of generated labels and the labeled frames can be reviewed in *Dataset View* and *Frame View* (D) to guide the refinement of definitions. *Training View* (E) shows the progress of TAL model training based on the generated labels.

6.2 Data Programming on Table Tennis Videos

6.2.1 Dataset Browsing and Frame Marking. Alex started with a dataset of 470 unlabeled table tennis video clips. The system first completed the extraction of the visual element information.

Video Browsing. Alex began by using the *Dataset View* (Figure 4) to get an overview of the videos. The *Dataset View* presents a cell matrix (Figure 4A), where each cell represents a video. By clicking on a cell, the video display module (Figure 4B) below displays the corresponding video. The timeline module (Figure 4C) includes a draggable progress bar to control the playback of the video and two parallel auxiliary timelines. Alex clicked on several videos to get a general sense of the dataset.

Drawing from his experience in table tennis data annotation, Alex believes that the serving action is distinct from other strokes because it "*involves a ball-throwing event.*" Therefore, he considered using this ball-throwing event as the blueprint for the key event definition. He pointed out that this key event could be break down into two states, "*when the ball is on top of the hand*" and "*when the ball is thrown into the air.*" To indicate that the ball is thrown, "*we could use a change in the relative direction of the ball and the player's head.*"

Key frame Marking. Using the frame marking functionality within the *Dataset View*, Alex marked two frames by clicking the button, and two markers were displayed on the timeline, as shown in Figure 4C1. These two frames represent the "ball held by hand" and "ball thrown into the air," respectively, which correspond to the two states for later reference. At this point, he noticed that in the table tennis broadcast videos, the visual features differ significantly when the serve player is oriented to the camera versus away from it. Alex proposed that two key events be defined, and he decided to "*first define the one for the serve action of players oriented to the camera.*"

6.2.2 Defining of Key Event. With the concept in mind, Alex proceeded with the defining. For convenience, we will refer to this key event as K_1 below.

Creation of Key Events and States. Alex initially created a new key event and initiated the editing process within the *Defining View*. Subsequently, within the *Defining View*, a timeline component (Figure 5A) was utilized for the purpose of managing the state of key events. Each node on the timeline represents a discrete state (Figure 5A1). Alex created two blank states, following Equation 3, and started editing the initial one.

Visual Element Manipulation. The visual elements involved in the state should be set according to Equation 6 and Equation 7. The *Defining View* supports two methods for visual element adding. The first method is by category, where users can select and add one visual element at a time to the canvas. The second method is through a hot start, allowing users to select a frame from any video

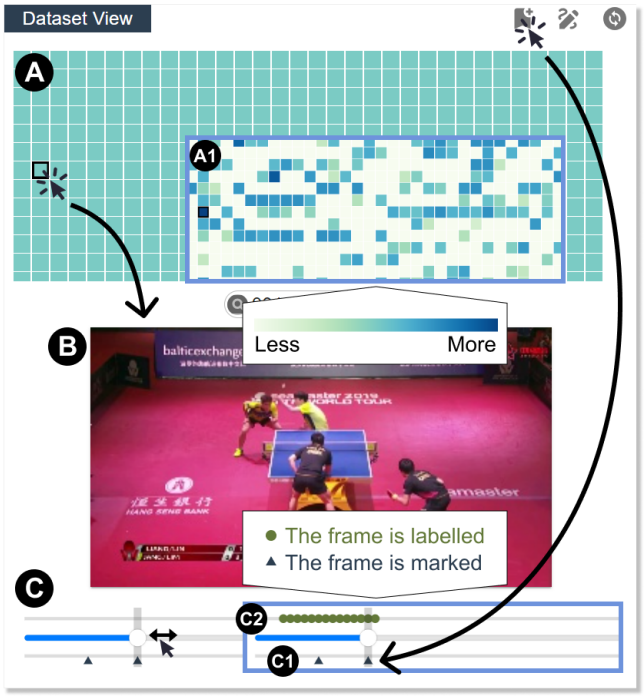

Figure 4: The *Dataset View* contains: (A) a cell matrix, where each cell represents a video, (B) a video display module, and (C) a timeline module containing two timelines, the top one (C2) showing the label distribution and the bottom one (C1) showing the user's markers.

and import all the extracted visual elements to the canvas based on their positions in the frame. Additionally, the selected frame can be set as the background of the canvas for reference. Each visual element, including objects and body parts, is represented as a node on the canvas, with the human skeleton also displayed (P3).

Alex remarked, "*Adding elements needed one by one is tedious. I've already marked some frames, so it'll be quicker to use those for a hot start.*" He then used the second method to add visual elements, locating the previously marked frame where "the player holds the ball and prepares to throw it up," and imported both the visual elements and the frame into the canvas. Alex then removed unnecessary elements, such as spectators. Since the nodes representing the hand and the ball were too close together, making them overlapping and difficult to select and link, Alex dragged the two nodes to adjust their position to separate them (P2, Figure 5B). It is noteworthy that the absolute position of visual elements is not a constraint and will not be considered in the final rules that generate training labels.

Constraint Setting. Alex then began setting the constraints between visual elements (P5, Equation 8, Equation 9). For state 1 of K_1, Alex explained, "*To capture the state where the ball is still in the hand and hasn't been thrown, there are two key relations: the contact between the ball and the hand and the direction of the ball relative to the head.*" He set the contact relation by clicking to link the ball and the hand (E5), with the relation visualized on the canvas (Figure 5E). For the direction relation, Alex created a valid direction range on the head node, visualized as a thick arc with the node at its center, and the arc's central angle representing the specified range. By dragging the arc, he adjusted its orientation (Figure 5C) and linked it with the ball node (Figure 5D), thereby establishing a direction constraint within a 70-degree interval toward the lower left (E5). Next, Alex established the direction relation between each player and the table. "*This relation is important,*" he noted, "*because in this key event, the player serving the ball should be positioned above the table, while the other player should be below it.*"

Alex then began defining state 2 of K_1. "*For state 2, I need to set the relation between the ball and the serve player's head,*" he explained. "*At this point, the ball is thrown up, positioned above and to the left of the center of the head.*" He configured this in the *Defining View*.

State Interval Setting. Referring to the previously marked frames, Alex set the time interval threshold (Equation 4) between the two states to 0.3 seconds on the timeline component (Figure 5A2).

6.2.3 Iterative Key Event Definition Refinement. At this point, Alex felt that his definition of K_1 had "*reached a temporary conclusion.*" He decided to "*check the quality of the labels first.*" After clicking the button, the system generated labels and displayed them in the *Dataset View* (P6).

Label Review. In the cell matrix component, each video cell is color-coded based on the number of labels (Figure 4A1). When viewing a video, the auxiliary timeline above the progress bar displays dots indicating the distribution of labels (Figure 4C2). Alex began by selecting a few cells to review the labeled frames in the corresponding videos. Concurrently, he utilized the *Frame View* to observe the retrieved frames that were based on the rules of the current state in *Defining View*. Alex noticed that "*most of the labels are correct, but there are some mislabeling and missing issues.*"

Iterative Modification. "*I want to see why this frame wasn't labeled,*" Alex remarked. He replaced the background in the canvas with the frame that wasn't retrieved and compared it with the previously defined relations on the canvas (Figure 5F). "*Ah, the range of the direction angle I set was a bit too narrow.*" He then adjusted the angle range to encompass the direction angles in several frames that should have met the conditions (P6, Figure 5G). Alex made several similar adjustments until he was "*basically satisfied*" with the results. "*I should address the mislabeling issue now,*" he said as he began reviewing frames that were incorrectly labeled. He discovered that some frames were mislabeled due to interference from other people in the video. In state 2, since only the direction of the ball relative to the head and the direction of the person relative to the table were constrained, those frames "*meet the rules when matching the person nearby.*" To filter out this issue, he added a pair of distance constraints.

Alex made several more modifications to improve the quality of the label. "*That's good enough,*" he said, deciding to stop making further adjustments. "*Even though the labels aren't completely accurate and some instances were still missed, from a data programming perspective, this is within a reasonable range.*" Similarly, Alex defined the key event, denoted as K_2, which pertains to the serve action of the player oriented away from the camera. Ultimately, Alex completed the label generation in 26.6 minutes.

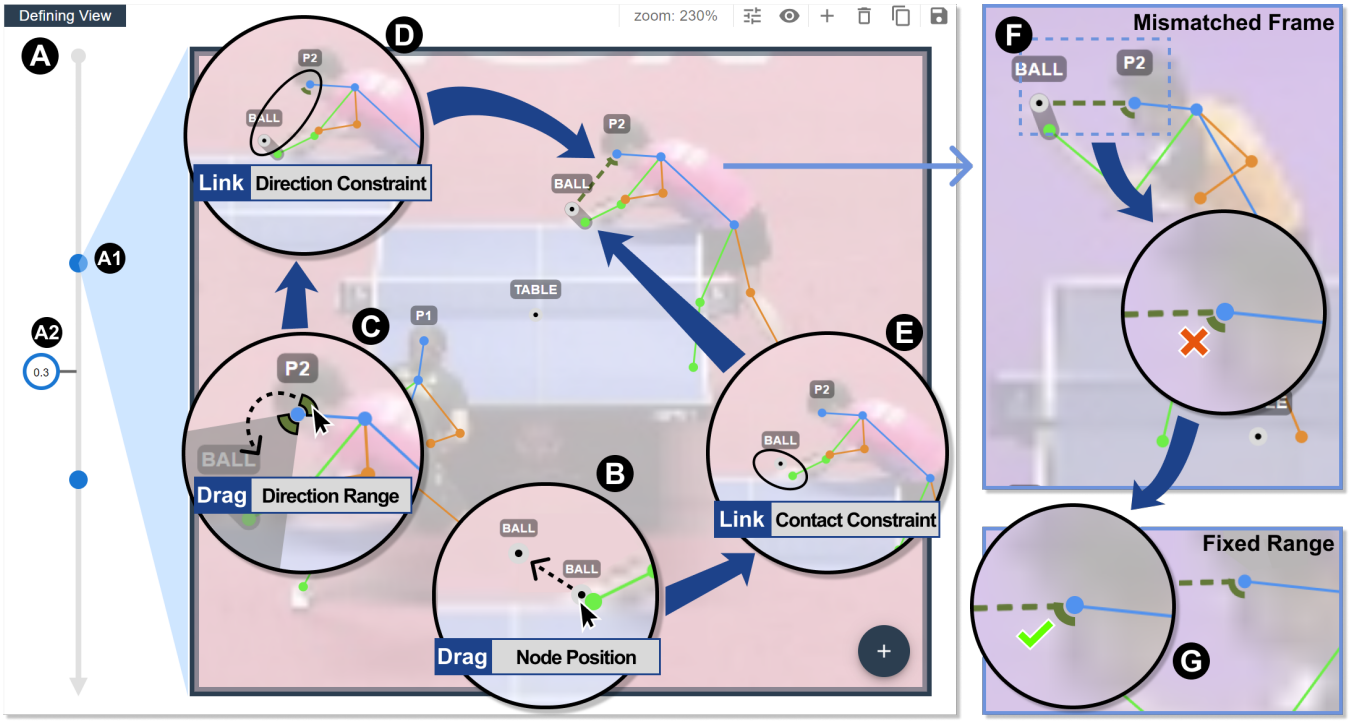

Figure 5: The *Defining View* contains: a timeline (A) for setting the number of states and time intervals, and a canvas featuring drag-and-link interactions. Users can drag to adjust node positions (B) and direction ranges (C), link nodes to define constraints such as direction (D), distance, and contact (E). This design also facilitates the refinement of key event definitions (F).

6.2.4 Final Training of the TAL Model. Alex clicks the button in the *Training View* (Figure 3E) to initiate TAL model training using the labels generated by the most recent version of the two defined key events.

Training Status. Alex observed the training process through the *Training View*, which illustrates the alterations in loss and mAP (calculated based on several labeled ground truths) as the number of training epochs increases. After the training converged, Alex acquired a TAL model for localizing serve actions and expressed satisfaction with the model performance.

6.3 Evaluation of the Framework

Comparative Study. We conducted a comparative study to evaluate the effectiveness of ProTAL by comparing the performance of a model built using ProTAL with models trained using traditional annotation-training workflows. We manually annotated Alex's videos for training (took a total of 15.7 hours to annotate) and an additional 130 videos for testing. First, a model was trained using SF-Net with full supervision labels, followed by another model trained with SF-Net using single-frame supervision labels. For single-frame labels, we selected the central frame of each action instance. Then these two models were compared with the model Alex built using ProTAL. As shown in Table 1, Alex's model significantly outperformed the single-frame supervised SF-Net in terms of average mAP and approached the performance of the model trained with full supervision. This outcome is impressive given that the labeling time was reduced by over 30 times. At IoU thresholds ranging from 0.3 to 0.7, Alex's model achieved higher mAP scores than the single-frame supervision model, showcasing its robustness in modeling action duration. These results demonstrate the effectiveness of ProTAL in constructing TAL models from unlabeled video dataset.

Additionally, ProTAL can be applied to various types of actions. Figure 6 presents screenshots that demonstrate the use of ProTAL to define key events for different actions, including single-human actions, human-human interactions, and human-object interactions.

7 User Study

The effectiveness of the framework in building TAL models from unlabeled video dataset was demonstrated in section 6. To further evaluate the design of the drag-and-link interactions, we conducted a comparative user study[2]. This study compared the drag-and-link interface of the prototype system with a baseline system that uses a form-based interface, which can be considered a version of ProTAL without the drag-and-link feature. It aimed to answer the following two research questions:

- Can drag-and-link interactions reduce the time consumed in defining key events (improve the efficiency of TAL data programming)?
- Can drag-and-link interactions reduce the number of iterations to refine key events (help define key events accurately)?

[2]The study has been approved by State Key Lab of CAD&CG, Zhejiang University.

Table 1: Model performance comparison with fully supervised method and single-frame supervised method.

	mAP@tIoU							avg-mAP
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.1:0.7
SF-Net w/ full label	1.000	0.992	0.953	0.897	0.834	0.663	0.494	0.833
SF-Net w/ single-frame label	0.982	0.909	0.836	0.767	0.613	0.327	0.116	0.650
ProTAL	0.909	0.909	0.892	0.888	0.854	0.728	0.596	0.825

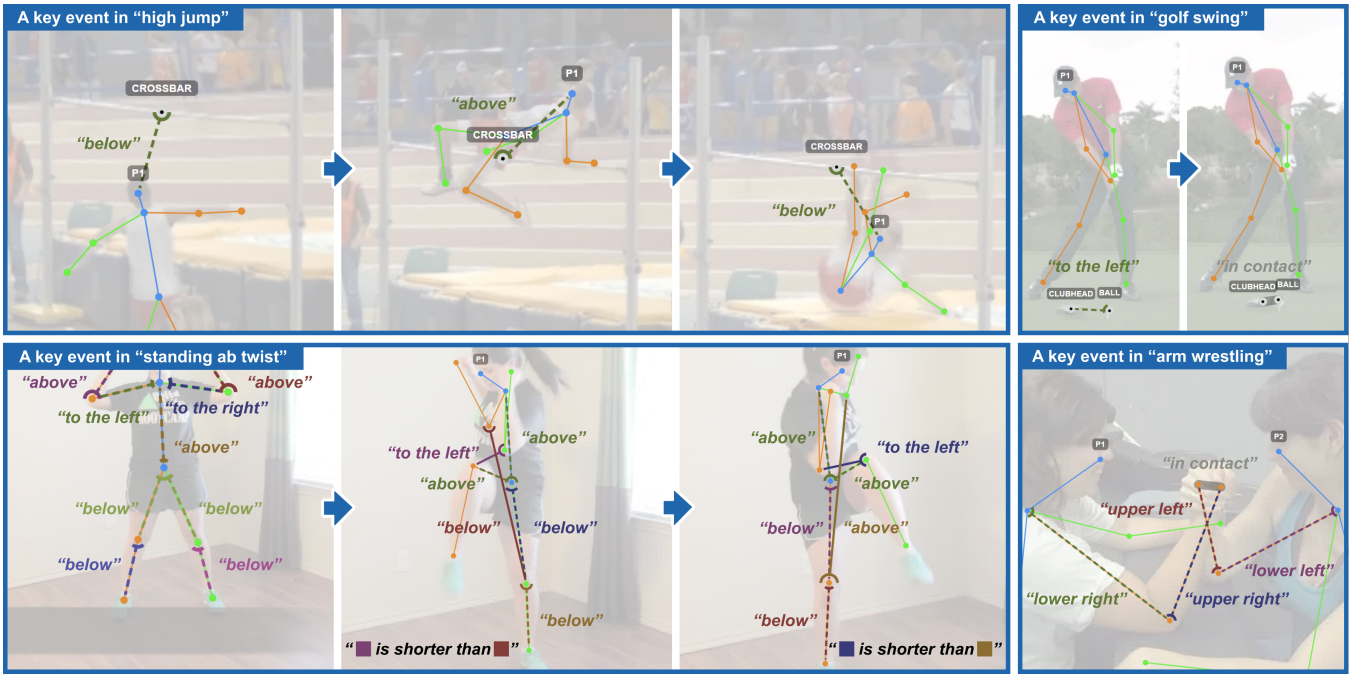

Figure 6: More usage examples. For "high jump," users can consider the direction between the head and the crossbar; for "golf swing," users can examine the direction and contact relation between the clubhead and the ball; and for "arm wrestling" and "standing ab twist," users can focus on the direction relation and relative distance between the joints.

7.1 Participants

A total of 12 action annotators (A1–A12, Age: 22–28) were recruited for the study, comprising both male and female participants. The participants are data analysts for various sports, with extensive expertise in annotating action data for purposes including quantitative analysis, visualization, and model training. Specifically, for model training purposes, eight participants had previously engaged in annotation for more than three projects, while four had engaged in at least one. All participants understood the TAL task setting and the deep learning-based TAL model training workflow, enabling them to provide valuable insights into the framework and system. They had no involvement in the preceding formative studies. For the subsequent tasks, the participants were randomly divided into two groups (G1 and G2), with six participants in each group. The study was conducted on a PC with a 32-inch monitor in the laboratory, and each participant received a compensation of $15.

7.2 Procedures

To assess how effectively ProTAL helps users translate abstract key event concepts into accurate, rule-based definitions, we compared it with a baseline system. Unlike ProTAL's drag-and-link interface, the baseline employs a form-based interface for key event definition, as shown in Figure 8. In this interface, each row represents a constraint, structured as a 5-tuple: (Element A, Element B, Relation Type, Parameter 1, Parameter 2). Users set constraints by selecting visual elements and relation types from drop-down menus and inputting relevant parameters. For instance, to define a direction relation such as "the angle of Element A relative to Element B is between 60 and 120 degrees", the users select "Element B," "Element A," and "direction," then specifies the angle range by entering the numerical values "60" and "120."

We designed four tasks, each involving the defining of key events in a floor exercise action (either "turns" or "tumbling") using one of the two systems (ProTAL or baseline). So, the four tasks were: (1) **ProTAL-turns**, (2) **baseline-turns**, (3) **ProTAL-tumbling**, and (4)

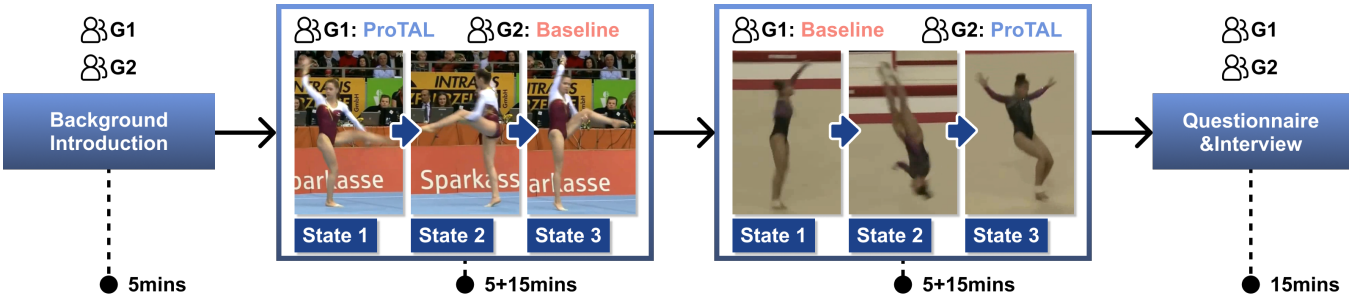

Figure 7: User study procedure. The user study was conducted in three phases. The initial phase comprised a background introduction. The subsequent phase involved the completion of two tasks, the first of which defined key events of "turns" and the second of which defined key events of "tumbling" in floor exercise. Participants were required to complete the tasks using different systems according to their group. Finally, a post-task questionnaire and an interview were conducted.

Figure 8: The baseline system employs a form-based interface for key event definition.

baseline-tumbling. These two single-human actions were selected for two reasons: first, to reduce the effort required to understand the key events so participants can focus on the interaction and second, to ensure a comparable level of complexity in defining key events for both actions. Moreover, since the interactions designed for human- and object-related visual elements are identical, the study results are expected to remain consistent across different action types. Group G1 was asked to complete tasks 1 and 4 in sequence, while group G2 was assigned tasks 2 and 3 in sequence. This ensured that each participant experienced both systems and different actions for each system. This was necessary because defining the same key event with another system would introduce bias. Additionally, the system order was alternated between groups to maintain fairness in comparison. The study for each participant was comprised of three phases (Figure 7):

Phase 1. Background Introduction (5mins). The first phase involved introducing key concepts to explain how data programming works for TAL, ensuring that participants developed a comprehensive understanding of the key event and the distinction between traditional data annotation and the data programming paradigm, thus preparing them for the tasks ahead.

Phase 2. Two Tasks (40mins). Each participant was required to complete two tasks in sequence. Before each task, we introduced the system and the action involved in the task. Participants were shown an video collected from FineGym [51] containing at least two instances of the action, with the action locations marked in the timeline. To minimize the impact of varying levels of participants' familiarity with the actions, we simplified the data programming task. First, participants familiarized themselves with the action in the video, after which we provided a general description of the key event directly. For "turns" action, the two instances in the video involved the athlete lifting her left leg to a near-horizontal position while rotating her body. For the "tumbling" task, the key event was the change in the athlete's torso direction. Participants were then asked to define the key events and generate labels on the given video using the assigned system. Each task was considered complete when the generated labels meets a specified quality (accuracy ≥ 0.8 and recall ≥ 0.2). To ensure balanced effort, the completion time (≤ 15 minutes) and the number of iterations (≤ 5) were capped to prevent participants from overthinking or defining key events too casually.

Phase 3. Post-task Questionnaire and Interview (15mins). After completing the two tasks, participants were asked to fill out a questionnaire to rate their experiences with the two systems. Following this, an interview was conducted to gather detailed user feedback on the two systems.

7.3 Research Data Collection and Analysis

To address the formulated research questions, a diverse set of data was collected for analysis, encompassing both quantitative and qualitative, as well as subjective and objective measures. Subjective data included questionnaire responses and interviews from the 12 participants.

Questionnaire. The questionnaire comprised two main sections. The first section focused on a comparative evaluation of the two

systems. To explore whether the drag-and-link interaction enhances usability, six key aspects were derived from the ten questions in the System Usability Scale (SUS). Participants rated these aspects on a comparative 7-point scale ranging from "*System A (baseline) much better*" to "no difference" to "*System B (our system) much better*." The aspects included: *overall performance*, reflecting the user's overall experience with the system; *easy to use*, indicating the ease of use of the system; *intuitive*, assessing functional consistency and learning intuitiveness; *cognitive effort*, measuring the cognitive load imposed on the users by the system; *physical effort*, evaluating the physical burden during operation; *practicality*, determining whether the system meets the user's practical needs. The second section required participants to rate specific interaction designs in our system, focusing on node (visual element) manipulation and constraint setting. These ratings were collected using a 7-point Likert scale. All responses were gathered for subsequent statistical analysis.

Interview. The interview focused on three topics: 1) the strengths and weaknesses of the two systems; 2) the underlying reasons behind participants' behaviors that differed from others during the tasks; and 3) suggestions for improving the system and framework. All interviews were recorded and transcribed for analysis. Feedback was categorized according to the research questions and interview topics, then reviewed and discussed by three coauthors. Key results were subsequently summarized.

For objective data, we recorded the entire process of the 12 participants performing the tasks and extracted relevant data metrics for analysis. Given that the label quality was required to meet predefined standards, the metrics analyzed focused on two aspects: task completion time and the number of iterations required to complete each task. These two metrics correspond to the two research questions.

Completion Time. To analyze task completion times, we used a paired t-test to compare the differences in completion times for the tasks completed on the two systems by the same group of participants. First, we ran a Shapiro-Wilk test at a significance level of 0.05 to check the normality of the paired differences. Given that the differences followed a normal distribution, we calculated the mean and standard deviation for both sets, along with the t-value and p-value for the comparison.

Number of Iterations. The number of iterations refers to the total number of times users generated labels based on the defined key events and review labels for refinement until the required label quality was achieved. The analysis for the number of iterations followed the same procedure as that used for task completion time.

7.4 Results and Feedback

7.4.1 Drag-and-link interaction enhances usability compared to the baseline system. Participants' ratings of the overall comparison between two systems are presented in Figure 9C. For overall experience, 83.3% (10/12) of participants preferred our system over the baseline, while only 16.7% (2/12) rated the baseline as "slightly better". Regarding interaction and visual encoding, 10 participants found our system "easier to use" and "more intuitive," whereas 2 participants, A4 and A10, who were highly familiar with gymnastics and the given actions, confidently defined the key events and found the baseline smooth to use. A4 noted that since he already knew exactly how to define the key event and what angle to set, he did not need the drag-and-link feature.

In terms of cognitive effort, 83.3% (10/12) of participants felt that our system required less cognitive effort. A1 and A12 noted that translating the desired direction into a numerical representation with the baseline was cognitively demanding. A11 observed that using the baseline often resulted in setting angles "*based on intuition and not sure.*" A7 emphasized that handling complex actions would be challenging with the baseline. Conversely, A2 and A4 mentioned that with a deep understanding of the action, the baseline could also be used effectively. Regarding physical effort, there was no clear preference between the two systems. A2 noted that the baseline interaction was also straightforward, the primary limitation of it is its lack of ability to facilitate interactive exploration. For practicality, all participants expressed a preference for using our system in their practical workflows compared to the baseline. These findings demonstrate that our system is more usable than the baseline, due to the introduction of the drag-and-link interaction.

7.4.2 The interaction design for node manipulation and constraint setting are intuitive. Most participants provided positive feedback on the manipulation of nodes representing human- and object-related visual elements, as shown in Figure 9D. However, there was a neutral rating regarding ease of use, with A2 suggesting, "*When dragging the entire human skeleton as a whole, it is easier to use and understand using a key combination to distinguish it from dragging individual nodes.*" Regarding constraint setting, most participants also gave favorable ratings for the interaction design. A1 highlighted the usefulness of the mode switch feature (Figure 3C1), noting that when the angle range is narrow, such as 15 degrees, the arc representing the range is small, making it somewhat challenging to drag. However, by switching the display mode, this issue was effectively resolved. Additionally, all participants found that setting a frame as the canvas background was highly useful, as it shows placement of visual elements and provided intuitive cues for setting constraints.

7.4.3 Drag-and-link interaction enhances efficiency in TAL data programming. Regarding the first research question, a paired t-test comparing task completion times between the two systems revealed that participants completed tasks faster using our system compared to baseline ($t = 3.04$, $p < 0.05$). The average completion time with our system was 518.8s ($SD = 172.8$), whereas the baseline required 622.1s ($SD = 249.4$), shown as Figure 9A. These results demonstrate that the drag-and-link interaction improves efficiency in TAL data programming. Additionally, both times were well within the allotted 15 minutes (900s), indicating that participants were able to understand and adapt to the data programming workflow and the constraint-setting logic with ease. Furthermore, All participants highlighted our system's ability to directly compare the defined constraints with a selected frame, which allows them to modify the constraints more efficiently to filter or retrieve some frames. A1 and A3 emphasized that the constraint copy feature also speeds up the process, noting that constraints are often similar between states.

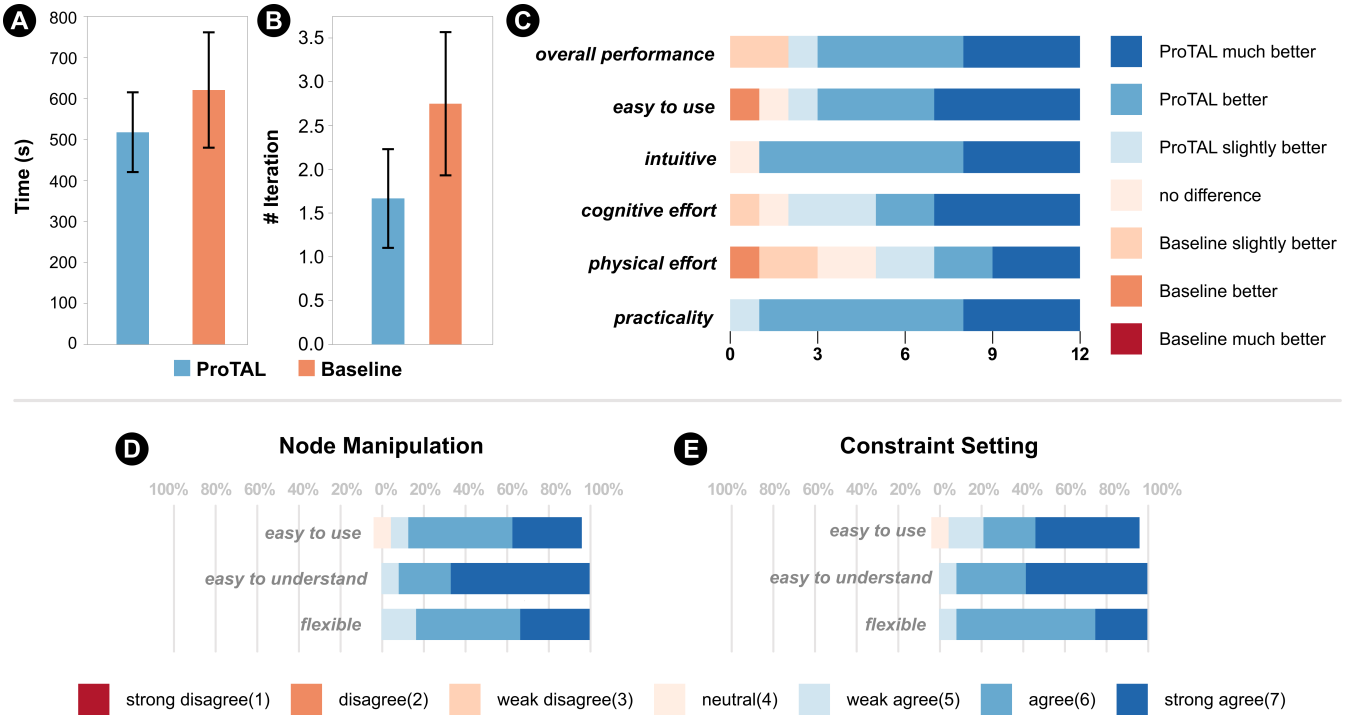

Figure 9: Quantitative results. (A) and (B) show the mean values of task completion time and the number of iterations, respectively, with error bars indicating the 95% confidence interval. (C) presents users' overall comparative ratings of our system versus the baseline. (D) and (E) display ratings for the interaction design of node manipulation and constraint setting in our system, respectively.

7.4.4 Drag-and-link interaction helps define key events accurately. The paired t-test revealed a significant reduction in the number of iterations required when using our system compared to baseline ($t = 2.60, p < 0.05$). In this study, the maximum number of iterations was capped at 5, with values ranging from 1 to 5. The average number of iterations for tasks completed with our system was 1.7 ($SD = 0.9$), compared to 2.8 ($SD = 1.3$) with the baseline, as shown in Figure 9B. These results indicate that participants required fewer iterations to complete the key event definitions using our system than with the baseline. With our system, 50% (6/12) of the participants completed the task in a single iteration, while only 25% (3/12) achieved this using the baseline. This suggests that the drag-and-link interaction enables users to define more accurate rules in the initial iteration. Furthermore, excluding cases with only one iteration, the average number of iterations with our system was 2.3, compared to 3.3 for the baseline, highlighting that the drag-and-link interaction facilitates more precise rule modifications. These findings show that the drag-and-link interface provides a more accurate approach to define key events, answering the second research question.

7.4.5 Individuals displayed a diversity of patterns of behavior and cognitive processes. During the tasks, notable diversity was observed in the way participants defined key events. For example, in terms of constraint setting, A1 and A2 initially set a wide angle range for direction constraints and then narrowed it in subsequent iterations. In contrast, A3 and A6 took the opposite approach. A1 indicated a preference for initially relaxing the constraints and then tightening them to eliminate incorrect frames, while A2 emphasized the importance of ensuring a high recall rate at the beginning. In contrast, A3 and A6 prioritized accuracy and then sought to improve recall. From the perspective of visual element selection, A6 and A12 found and attempted to define different versions of the key event definitions for the tumbling action. They focused on the direction of the person's feet and head, defining more complex but effective key events. This phenomenon is consistent with the nature of key events, where different users may have different interpretations of the same action. For any action, there may be several reasonable key events to define.

7.4.6 The system exhibits significant potential for improvement. During the interviews, participants agreed that ProTAL offers a promising solution to the high cost of action annotation and provided several suggestions for improvement to address its perceived weaknesses and improve usability. Three participants expressed concern about the numerical accuracy of the constraints, especially since they had previously annotated precise data. They recommended combining drag-and-link interaction with direct numerical control in the baseline system to enhance numerical accuracy. A8 suggested implementing an adsorption effect to adjust the direction range to improve interaction efficiency and precision. Currently,

our system highlights the generated labels in the *Dataset View*, allowing users to review and adjust constraints for mislabeled frames. A1 suggested further refinements, such as highlighting specific regions within the mislabeled frames that do not meet the defined constraints. This feature would eliminate the need for users to manually identify which constraints caused errors. A7 recommended that the system automatically update constraints based on mislabeled frames. In addition, A1, A7, and A8 suggested that the integration of language models could significantly improve the efficiency of labeling by recommending potential constraint candidates when defining key events.

8 Discussion

In this section, we reflect on our interactive video programming framework and the prototype system, summarizing the implications we learned. We also discuss the feasibility of ProTAL, explore possible future research directions, and outline the limitations of current research based on user feedback and observations.

8.1 Implications for Designing Video Programming Framework

The effectiveness of ProTAL and the usability of the prototype system are demonstrated, highlighting the potential to inspire the design of data programming frameworks for other video tasks.

- **Identify the appropriate constraint space for new video programming tasks**. In this paper, our goal is to develop a video programming framework for TAL. We began by decomposing actions into finer-grained key events, defining them through changes in the relations between visual elements, which serve as labeling functions in data programming. To better understand the constraints involved in defining key events, we conducted a workshop study that led to the derivation of the constraint space. This space guided the implementation of the prototype system, which was successfully applied to practical scenarios. However, this constraint space may not encompass all video tasks. When designing video programming frameworks for other tasks, it is crucial to carefully derive the constraint space for them. For instance, when developing a system for higher-level event recognition, such as tactical analysis in team sports [35], the constraint spaces should be extended to encompass lineup information, player roles, etc.
- **Decomposing and simplifying data programming objects for highly complex tasks**. Data programming is being applied to increasingly complex tasks and data, moving from text to images and from video classification to action localization. However, the complexity that rules can handle is not keeping pace with the growing complexity of tasks and data. In addition, the rules must remain simple enough, as overly complex rules would make direct annotation more efficient than data programming. Therefore, when extending video programming to more complex tasks, it is essential to decompose complicated programming objects, such as decomposing actions into key events with a simpler structure and programming key events. Such decomposed objects can be defined by rules of manageable complexity, facilitating data programming. Furthermore, advanced models are needed to use these weak labels for effective model training.

8.2 Potential of ProTAL

We reflect on the design and potential of the framework and system, focusing on adaptability and scalability.

- **Adaptability to broader applications.** Although ProTAL is specifically designed for TAL, its drag-and-link interaction design, along with its visual encoding of visual elements, human poses, and constraints between them, can be extended to other tasks involving actions or interactive events, such as action quality scoring and spatial action segmentation.
- **Efficiency in handling larger datasets.** ProTAL effectively scales with dataset size without increasing annotator workload. The annotator's time cost remains consistent as the dataset size grows, since they only need to define key events. ProTAL then automatically applies these definitions to match all frames, eliminating the need for additional manual intervention. This efficiency makes ProTAL a viable tool for large-scale video datasets.
- **Extension to higher-dimensional scenarios.** Although currently focused on video data, ProTAL can be expanded to handle 3D [7] or even 4D scenarios, such as those in virtual reality [36, 75] and motion capture systems. By incorporating 3D detection or tracking modules, the system's canvas can be extended to define key events in the 3D space. This extension opens opportunities for annotating complex interactions and actions within immersive environments.

8.3 Limitations & Future Work

8.3.1 Current limitations of ProTAL. While ProTAL has proven effective for temporal annotation across various types of actions, it may face challenges in complex in-the-wild scenarios. Firstly, dense and overlapping objects in videos can complicate the recognition and extraction of visual elements, thereby disrupting the data programming workflow. Changes in viewpoint present another challenge. In cases where the video dataset features distinct viewpoints, such as the two viewpoints in the table tennis match videos discussed in section 6, users can define separate key events for each viewpoint. However, dynamic or excessively varied viewpoints may require viewpoint alignment or defining key events within a 3D environment to ensure consistency. Additionally, videos shot from a first-person perspective introduce unique complexities, such as handling the hands, body, or other visible parts of the shooter, which may require tailored approaches. Further exploration will be conducted to address these limitations.

8.3.2 Future work. Moreover, there are several opportunities for future work:

- **Expanding the space of constraints for greater flexibility**. Currently, ProTAL provides a set of constraints based on the relations between visual elements. However, in order to distinguish actions in a more fine-grained way, such as distinguishing between tumbling actions on the ground and in the air, ProTAL needs to support a larger constraint space. This extension would allow users to define more nuanced actions and handle complicated action variations, further improving the accuracy and flexibility of action annotation.

- **Domain knowledge-driven constraint recommendation**. Our user study has shown that users may have different cognitive understandings of actions, and users who have a deeper understanding of specific actions can define the key event more efficiently. Therefore, ProTAL can benefit from integrating domain knowledge to automatically recommend appropriate constraints based on the specific action. By integrating expertise in different actions, ProTAL can guide users to select constraints that are more appropriate for their tasks, thus reducing cognitive load and improving the accuracy of the annotation process.
- **Integration with large multimodal models**. Incorporating large multimodal models into ProTAL could enable more advanced AI-powered features. Using video, image, and text data, multimodal models could automatically suggest key events and constraints based on the context of the action, simplifying the process of defining key events. Furthermore, large multimodal models offer the potential to integrate ProTAL's visual element extraction and rule-based frame matching steps. For example, users could enter rules in natural language along with a frame, and the models could determine whether the frame satisfies those rules, further increasing flexibility.

9 Conclusion

We present ProTAL, a novel video programming framework designed for TAL. The framework addresses the significant challenge of decomposing actions into meaningful substructures by decomposing actions into key events that are easier to define and recognize. ProTAL then presents a drag-and-link interaction design that allows users to define key events through intuitive interactions. These key event definitions, which constrain relations between visual elements, serve as data programming rules that generate frame-wise action labels for large-scale unlabeled videos. With these labels, a semi-supervised method is used to effectively train TAL models.

Based on the proposed framework, a system was implemented. The effectiveness and usability of the implemented system in TAL annotation and training was demonstrated through a practical usage scenario and a user study. Feedback from participants highlighted the design of the drag-and-link interaction. These results also provide valuable guidance for the development of future video programming frameworks.

Acknowledgments

This work was supported by NSFC (U22A2032) and Key Scientific Research Project of the Department of Education of Guangdong Province (2024ZDZX3012). The author also gratefully acknowledges the support of Zhejiang University Education Foundation Qizhen Scholar Foundation.

References

[1] Maya Antoun and Daniel Asmar. 2023. Human object interaction detection: Design and survey. *Image and Vision Computing* 130, C (2023), 104617. https://doi.org/10.1016/J.IMAVIS.2022.104617

[2] Stephen H. Bach, Bryan Dawei He, Alexander Ratner, and Christopher Ré. 2017. Learning the Structure of Generative Models without Labeled Data. In *Proceedings of the 34th International Conference on Machine Learning*. 273–282.

[3] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. 2020. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications* 79, 41-42 (2020), 30509–30555. https://doi.org/10.1007/S11042-020-09004-3

[4] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4724–4733. https://doi.org/10.1109/CVPR.2017.502

[5] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1130–1139. https://doi.org/10.1109/CVPR.2018.00124

[6] Changjian Chen, Jiashu Chen, Weikai Yang, Haoze Wang, Johannes Knittel, Xibin Zhao, Steffen Koch, Thomas Ertl, and Shixia Liu. 2024. Enhancing Single-Frame Supervision for Better Temporal Action Localization. *IEEE Transactions on Visualization and Computer Graphics* 30, 6 (2024), 2903–2915. https://doi.org/10.1109/TVCG.2024.3388521

[7] Lu Chen, Sida Peng, and Xiaowei Zhou. 2021. Towards efficient and photorealistic 3D human reconstruction: A brief survey. *Visual Informatics* 5, 4 (2021), 11–19. https://doi.org/10.1016/j.visinf.2021.10.003

[8] Dongjin Choi, Sara Evensen, Çagatay Demiralp, and Estevam Hruschka. 2021. TagRuler: Interactive Tool for Span-Level Data Programming by Demonstration. In *Companion Proceedings of the Web Conference 2021*. 673–677. https://doi.org/10.1145/3442442.3458602

[9] L. Minh Dang, Kyungbok Min, Hanxiang Wang, Md. Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. 2020. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition* 108 (2020), 107561. https://doi.org/10.1016/J.PATCOG.2020.107561

[10] Dazhen Deng, Jiang Wu, Jiachen Wang, Yihong Wu, Xiao Xie, Zheng Zhou, Hui Zhang, Xiaolong (Luke) Zhang, and Yingcai Wu. 2021. EventAnchor: Reducing Human Interactions in Event Annotation of Racket Sports Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 73:1–73:13. https://doi.org/10.1145/3411764.3445431

[11] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. 2022. Revisiting Skeleton-based Action Recognition. In *Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2959–2968. https://doi.org/10.1109/CVPR52688.2022.00298

[12] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. 2016. DAPs: Deep Action Proposals for Action Understanding. In *Proceedings of Computer Vision – ECCV 2016*. 768–784. https://doi.org/10.1007/978-3-319-46487-9_47

[13] Sara Evensen, Chang Ge, and Çagatay Demiralp. 2020. Ruler: Data Programming by Demonstration for Document Labeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 1996–2005. https://doi.org/10.18653/V1/2020.FINDINGS-EMNLP.181

[14] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. 2015. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 961–970.

[15] Gueter Josmy Faure, Min-Hung Chen, and Shang-Hong Lai. 2023. Holistic Interaction Transformer Network for Action Detection. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 3329–3339. https://doi.org/10.1109/WACV56688.2023.00334

[16] Yutong Feng, Jianwen Jiang, Ziyuan Huang, Zhiwu Qing, Xiang Wang, Shiwei Zhang, Mingqian Tang, and Yue Gao. 2021. Relation Modeling in Spatio-Temporal Action Localization.

[17] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. 2017. Cascaded Boundary Regression for Temporal Action Detection. In *Proceedings of British Machine Vision Conference 2017*.

[18] Georgia Gkioxari, Ross B. Girshick, Piotr Dollár, and Kaiming He. 2018. Detecting and Recognizing Human-Object Interactions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8359–8367. https://doi.org/10.1109/CVPR.2018.00872

[19] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. 2018. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6047–6056. https://doi.org/10.1109/CVPR.2018.00633

[20] Dongming Han, Jiacheng Pan, Xiaodong Zhao, and Wei Chen. 2021. NetV.js: A web-based library for high-efficiency visualization of large-scale graphs and networks. *Visual Informatics* 5, 1 (2021), 61–66. https://doi.org/10.1016/j.visinf.2021.01.002

[21] Jianben He, Xingbo Wang, Kam Kwai Wong, Xijie Huang, Changjian Chen, Zixin Chen, Fengjie Wang, Min Zhu, and Huamin Qu. 2024. VideoPro: A Visual Analytics Approach for Interactive Video Programming. 30, 1 (2024), 87–97. https://doi.org/10.1109/TVCG.2023.3326586

[22] Md Naimul Hoque, Wenbin He, Arvind Kumar Shekar, Liang Gou, and Liu Ren. 2023. Visual concept programming: A visual analytics approach to injecting human intelligence at scale. *IEEE Transactions on Visualization and Computer*

Graphics 29, 1 (2023), 74–83. https://doi.org/10.1109/TVCG.2022.3209466

[23] Edwin L Hutchins, James D Hollan, and Donald A Norman. 1985. Direct manipulation interfaces. *Human–computer interaction* 1, 4 (1985), 311–338.

[24] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. 2023. RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose. https://doi.org/10.48550/ARXIV.2303.07399

[25] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. 2014. "THUMOS Challenge: Action Recognition with a Large Number of Classes". http://crcv.ucf.edu/THUMOS14/.

[26] Pushpajit Khaire and Praveen Kumar. 2022. Deep learning and RGB-D based human action, human-human and human-object interaction recognition: A survey. *Journal of Visual Communication and Image Representation* 86, C (2022), 103531. https://doi.org/10.1016/J.JVCIR.2022.103531

[27] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J. Kim. 2021. HOTR: End-to-End Human-Object Interaction Detection with Transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 74–83. https://doi.org/10.1109/CVPR46437.2021.00014

[28] Kuno Kurzhals, Marcel Hlawatsch, Christof Seeger, and Daniel Weiskopf. 2017. Visual Analytics for Mobile Eye Tracking. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 301–310. https://doi.org/10.1109/TVCG.2016.2598695

[29] Zhe Li, Yazan Abu Farha, and Jurgen Gall. 2021. Temporal Action Segmentation From Timestamp Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8365–8374. https://doi.org/10.1109/CVPR46437.2021.00826

[30] Zhengyang Li, Jie Li, and Xinying Ma. 2025. Representing multi-dimensional data as graph to visualize and analyze subset communities. *Journal of Visualization* (2025). https://doi.org/10.1007/s12650-025-01045-w

[31] Qinying Liu, Zilei Wang, and Shenghai Rong. 2023. Improve Temporal Action Proposals using Hierarchical Context. *Pattern Recognition* 140 (2023), 109560. https://doi.org/10.1016/j.patcog.2023.109560

[32] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2023. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499* (2023).

[33] Shuming Liu, Chen-Lin Zhang, Chen Zhao, and Bernard Ghanem. 2023. End-to-End Temporal Action Detection with 1B Parameters Across 1000 Frames. *arXiv preprint arXiv: 2311.17241* (2023), 18591–18601.

[34] Yuan Liu, Jingyuan Chen, Zhenfang Chen, Bing Deng, Jianqiang Huang, and Hanwang Zhang. 2021. The Blessings of Unlabeled Background in Untrimmed Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6176–6185. https://doi.org/10.1109/CVPR46437.2021.00611

[35] Ziao Liu, Xiao Xie, Moqi He, Wenshuo Zhao, Yihong Wu, Liqi Cheng, Hui Zhang, and Yingcai Wu. 2024. Smartboard: Visual Exploration of Team Tactics with LLM Agent. *IEEE Transactions on Visualization and Computer Graphics* (2024), 1–11. https://doi.org/10.1109/TVCG.2024.3456200

[36] Júlio Castro Lopes and Rui Pedro Lopes. 2024. Computer Vision in Augmented, Virtual, Mixed and Extended Reality environments—A bibliometric review. *Visual Informatics* 8, 4 (2024), 13–22. https://doi.org/10.1016/j.visinf.2024.11.002

[37] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. 2020. SF-Net: Single-Frame Supervision for Temporal Action Localization. In *Proceedings of Computer Vision - ECCV 2020 - 16th European Conference*. 420–437. https://doi.org/10.1007/978-3-030-58548-8_25

[38] Ayana Murakami and Takayuki Itoh. 2025. Flexible optimization of hierarchical graph layout by genetic algorithm with various conditions. *Journal of Visualization* 28, 1 (2025), 181–204. https://doi.org/10.1007/s12650-024-01018-5

[39] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. 2022. Semi-supervised Temporal Action Detection with Proposal-Free Masking. In *Proceedings of Computer Vision – ECCV 2022*. 663–680. https://doi.org/10.1007/978-3-031-20062-5_38

[40] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. 2018. Weakly Supervised Action Localization by Sparse Temporal Pooling Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6752–6761. https://doi.org/10.1109/CVPR.2018.00706

[41] Phuc Nguyen, Deva Ramanan, and Charless Fowlkes. 2019. Weakly-Supervised Action Localization With Background Modeling. In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 5501–5510. https://doi.org/10.1109/ICCV.2019.00560

[42] Jorge Piazentin Ono, Arvi Gjoka, Justin Salamon, Carlos A. Dietrich, and Cláudio T. Silva. 2019. HistoryTracker: Minimizing Human Interactions in Baseball Game Annotation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 63. https://doi.org/10.1145/3290605.3300293

[43] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. 2021. Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. 464–474. https://doi.org/10.1109/CVPR46437.2021.00053

[44] G. Paoletti, J. Cavazza, C. Beyan, and A. Del Bue. 2021. Subspace Clustering for Action Recognition with Covariance Representations and Temporal Pruning. In *2020 25th International Conference on Pattern Recognition (ICPR)*. 6035–6042. https://doi.org/10.1109/ICPR48806.2021.9412060

[45] Jathushan Rajasegaran, Georgios Pavlakos, Angjoo Kanazawa, Christoph Feichtenhofer, and Jitendra Malik. 2023. On the Benefits of 3D Pose and Tracking for Human Action Recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 640–649. https://doi.org/10.1109/CVPR52729.2023.00069

[46] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *Proceedings of the VLDB Endowment* 11, 3 (2017), 269–282. https://doi.org/10.14778/3157794.3157797

[47] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training Complex Models with Multi-Task Weak Supervision. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. 4763–4771. https://doi.org/10.1609/AAAI.V33I01.33014763

[48] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data Programming: Creating Large Training Sets, Quickly. In *Advances in Neural Information Processing Systems*, Vol. 29.

[49] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. 2024. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. arXiv:2401.14159 [cs.CV]

[50] Benjamin Renoust, Haolin Ren, and Guy Melançon. 2019. Animated Drag and Drop Interaction for Dynamic Multidimensional Graphs. *arXiv preprint arXiv: 1902.01564* (2019).

[51] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. 2020. FineGym: A Hierarchical Video Dataset for Fine-Grained Action Understanding. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2613–2622. https://doi.org/10.1109/CVPR42600.2020.00269

[52] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. 2023. Temporal Action Localization with Enhanced Instant Discriminability. *arXiv preprint arXiv: 2309.05590* (2023).

[53] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Lit, and Dacheng Tao. 2023. TriDet: Temporal Action Detection with Relative Boundary Modeling. In *Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 18857–18866. https://doi.org/10.1109/CVPR52729.2023.01808

[54] Zheng Shou, Dongang Wang, and Shih-Fu Chang. 2016. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1049–1058. https://doi.org/10.1109/CVPR.2016.119

[55] Alexandros Stergiou and Ronald Poppe. 2019. Analyzing human-human interactions: A survey. *Computer Vision and Image Understanding* 188, C (2019), 102799. https://doi.org/10.1016/J.CVIU.2019.102799

[56] Tan Tang, Yanhong Wu, Yingcai Wu, Lingyun Yu, and Yuhong Li. 2022. VideoModerator: A Risk-aware Framework for Multimodal Video Moderation in E-Commerce. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2022), 846–856. https://doi.org/10.1109/TVCG.2021.3114781

[57] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In *Advances in Neural Information Processing Systems*. 10078–10093.

[58] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV)*. 4489–4497. https://doi.org/10.1109/ICCV.2015.510

[59] Paroma Varma, Frederic Sala, Ann He, Alexander Ratner, and Christopher Ré. 2019. Learning Dependency Structures for Weak Supervision Models. In *Proceedings of the 36th International Conference on Machine Learning*. 6418–6427.

[60] Binglu Wang, Yongqiang Zhao, Le Yang, Teng Long, and Xuelong Li. 2024. Temporal Action Localization in the Deep Learning Era: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 4 (2024), 2171–2190. https://doi.org/10.1109/TPAMI.2023.3330794

[61] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. 2017. UntrimmedNets for Weakly Supervised Action Recognition and Detection. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6402–6411. https://doi.org/10.1109/CVPR.2017.678

[62] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2019. Temporal Segment Networks for Action Recognition in Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 11 (2019), 2740–2755. https://doi.org/10.1109/TPAMI.2018.2868668

[63] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. 2020. Learning Human-Object Interaction Detection Using Interaction Points. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4115–4124. https://doi.org/10.1109/CVPR42600.2020.00417

[64] Yanyan Wang, Zhanning Bai, Zhifeng Lin, Xiaoqing Dong, Yingchaojie Feng, Jiacheng Pan, and Wei Chen. 2021. G6: A web-based library for graph visualization. *Visual Informatics* 5, 4 (2021), 49–55. https://doi.org/10.1016/j.visinf.2021.12.003

[65] Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and Mohan S. Kankanhalli. 2019. Learning to Detect Human-Object Interactions With Knowledge. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019–2028. https://doi.org/10.1109/CVPR.2019.00212

[66] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. 2024. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 38571–38584.

[67] Katsu Yamane and Yoshihiko Nakamura. 2003. Natural Motion Animation through Constraining and Deconstraining at Will. *IEEE Trans. Vis. Comput. Graph.* 9, 3 (2003), 352–360. https://doi.org/10.1109/TVCG.2003.1207443

[68] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *Proceedings of AAAI Conference on Artificial Intelligence*, 7444–7452. https://doi.org/10.1609/aaai.v32i1.12328

[69] Le Yang, Junwei Han, Tao Zhao, Tianwei Lin, Dingwen Zhang, and Jianxin Chen. 2022. Background-Click Supervision for Temporal Action Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (2022), 9814–9829. https://doi.org/10.1109/TPAMI.2021.3132058

[70] Bangpeng Yao and Li Fei-Fei. 2010. Modeling mutual context of object and human pose in human-object interaction activities. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 17–24. https://doi.org/10.1109/CVPR.2010.5540235

[71] Runhao Zeng, Wenbing Huang, Chuang Gan, Mingkui Tan, Yu Rong, Peilin Zhao, and Junzhou Huang. 2019. Graph Convolutional Networks for Temporal Action Localization. In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 7093–7102. https://doi.org/10.1109/ICCV.2019.00719

[72] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, P. Zhao, Junzhou Huang, and Chuang Gan. 2022. Graph Convolutional Module for Temporal Action Localization in Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 10 (2022), 6209–6223. https://doi.org/10.1109/TPAMI.2021.3090167

[73] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2023. Data-centric Artificial Intelligence: A Survey. *arXiv preprint arXiv: 2303.10158* (2023).

[74] Chen-Lin Zhang, Jianxin Wu, and Yin Li. 2022. ActionFormer: Localizing Moments of Actions with Transformers. In *Proceedings of Computer Vision - ECCV 2022 - 17th European Conference*. 492–510. https://doi.org/10.1007/978-3-031-19772-7_29

[75] Yue Zhang, Zhenyuan Wang, Jinhui Zhang, Guihua Shan, and Dong Tian. 2023. A survey of immersive visualization: Focus on perception and interaction. *Visual Informatics* 7, 4 (2023), 22–35. https://doi.org/10.1016/j.visinf.2023.10.003

[76] Chen Zhao, Ali Thabet, and Bernard Ghanem. 2021. Video Self-Stitching Graph Network for Temporal Action Localization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 13638–13647. https://doi.org/10.1109/ICCV48922.2021.01340

[77] Chen Zhao, Ali K. Thabet, and Bernard Ghanem. 2021. Video Self-Stitching Graph Network for Temporal Action Localization. In *Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 13638–13647. https://doi.org/10.1109/ICCV48922.2021.01340

[78] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2017. Temporal Action Detection with Structured Segment Networks. In *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*. 2933–2942. https://doi.org/10.1109/ICCV.2017.317

[79] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2020. Temporal Action Detection with Structured Segment Networks. *International Journal of Computer Vision* 128, 1 (2020), 74–95. https://doi.org/10.1007/S11263-019-01211-2

[80] Qian Zhou, David Ledo, George Fitzmaurice, and Fraser Anderson. 2024. TimeTunnel: Integrating Spatial and Temporal Motion Editing for Character Animation in Virtual Reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 101:1–101:17. https://doi.org/10.1145/3613904.3641927

[81] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, and Jian Sun. 2021. End-to-End Human Object Interaction Detection With HOI Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11825–11834.