# *VideoPro*: A Visual Analytics Approach for Interactive Video Programming

Jianben He, Xingbo Wang, Kam Kwai Wong, Xijie Huang, Changjian Chen,
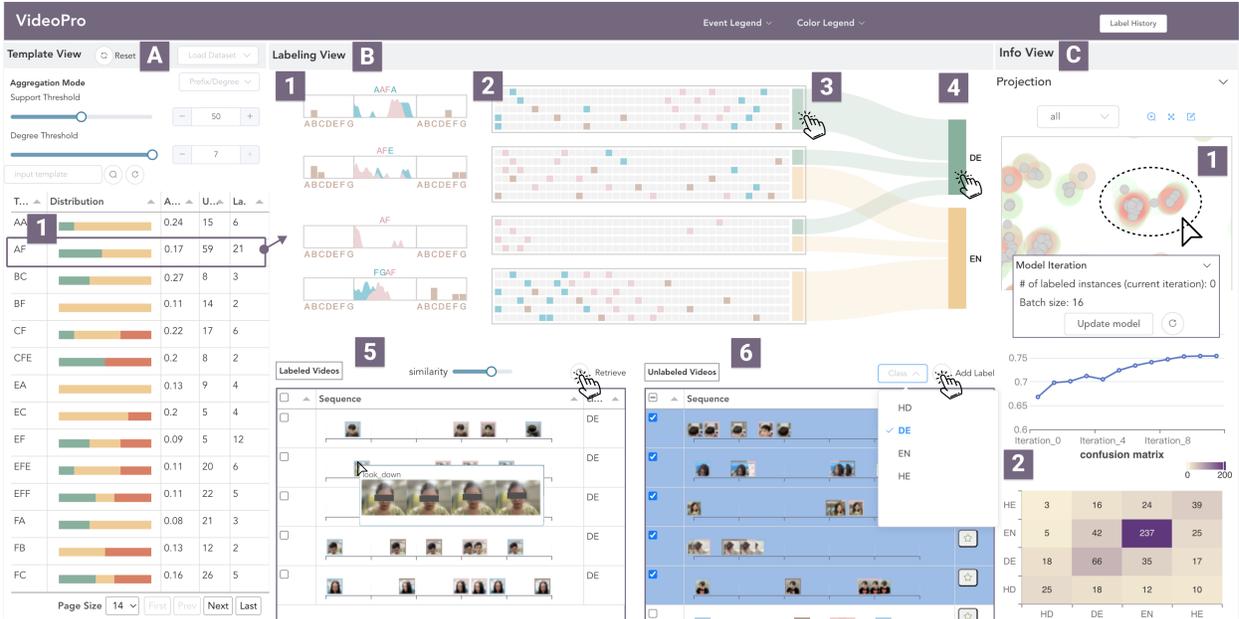Zixin Chen, Fengjie Wang, Min Zhu, and Huamin Qu

Fig. 1: The *VideoPro* interface consists of three major views. The *Template View* (A) offers descriptive statistics and rich interactions to facilitate multi-faceted exploration and comprehension of labeling templates. The *Labeling View* (B) provides a summary of the nuanced event compositions within the selected template to allow effective template validation and refinement. It also displays retrieved matching videos for efficient examination and at-scale programming. The *Info View* (C) presents comprehensive information regarding data embedding distribution in latent space and the model iteration process.

**Abstract**— Constructing supervised machine learning models for real-world video analysis require substantial labeled data, which is costly to acquire due to scarce domain expertise and laborious manual inspection. While data programming shows promise in generating labeled data at scale with user-defined labeling functions, the high dimensional and complex temporal information in videos poses additional challenges for effectively composing and evaluating labeling functions. In this paper, we propose *VideoPro*, a visual analytics approach to support flexible and scalable video data programming for model steering with reduced human effort. We first extract human-understandable events from videos using computer vision techniques and treat them as atomic components of labeling functions. We further propose a two-stage template mining algorithm that characterizes the sequential patterns of these events to serve as labeling function templates for efficient data labeling. The visual interface of *VideoPro* facilitates multifaceted exploration, examination, and application of the labeling templates, allowing for effective programming of video data at scale. Moreover, users can monitor the impact of programming on model performance and make informed adjustments during the iterative programming process. We demonstrate the efficiency and effectiveness of our approach with two case studies and expert interviews.

**Index Terms**—Interactive machine learning, data programming, video exploration and analysis

---◆---

## 1 INTRODUCTION

- *J. He, X. Wang, KK Wong, X. Huang, Z. Chen, H. Qu are with the Hong Kong University of Science and Technology, Hong Kong, China. X. Wang is the corresponding author. E-mail: {jhebt, xwangeg, kkwongar, xhuangbs, zchendf, huamin}@ust.hk.*
- *C. Chen is with the Tsinghua University, Beijing, China. Email: changjianchen.me@gmail.com*
- *F. Wang, M. Zhu are with the Sichuang University, Chengdu, China. Email: wangfengjie@stu.scu.edu.cn, and zhumin@scu.edu.cn*

The growing prevalence of video recordings has opened up opportunities for video analysis in numerous applications. For instance, sports analysts analyze athletic maneuvers from recorded competitions to enhance strategic decision-making [64, 73], while scientists study videotaped experiments to identify behavioral patterns and gather evidence to support their hypotheses [35, 58]. Recently, deep learning models have shown remarkable potential in automatically detecting domain-specific events in videos, significantly improving analysis efficiency over manual video review [20, 63]. However, building such models necessitates abundant labeled data, and the labeling process can be quite time-consuming and challenging, especially for complex video

content that needs specialized domain knowledge and expertise [60].

Data programming [38, 80] has emerged as a promising paradigm for reducing manual labeling efforts. By defining labeling functions based on their domain knowledge, users can assign weak-supervision labels to raw data for model training [53]. For example, for text labeling tasks, if a cluster of sentences contains similar harmful words, users can define a labeling function to assign a "toxic" label to the cluster [52]. For images, users can define a set of rules that assemble image segments (*e.g.*, head, body) to formulate new visual objects (*e.g.*, person) [26]. Nevertheless, compared with text and image data, programming video data is particularly challenging. First, it is demanding to decompose video data into meaningful semantic units for building labeling functions. Videos contain segments of events that involve complex interactions among multiple objects over time. Particularly, the temporal context information could largely influence the semantic meaning of the video content. For example, two cooking videos with the same food ingredients but different cooking steps can result in dishes with distinct textures and flavors. Therefore, labeling functions need to model the variations and nuances in temporal relationships among multiple events. However, manually constructing such functions is challenging, given the wide range of events and their complex temporal dependencies.

Second, evaluating, refining, and applying labeling functions for high-quality label generation and efficient model training is non-trivial. Multiple factors, including data coverage, model performance, and semantic meanings of labeling functions, need to be considered before applying them to large unlabeled video datasets. Furthermore, during the iterative programming process, users need to continuously monitor model performance under the impact of different labeling functions, and make corresponding refinement leveraging their domain knowledge. Developing an effective tool to facilitate and expedite the programming process with minimal user efforts is also challenging.

To solve the above challenges, we introduce *VideoPro*, a visual analytics approach that enables flexible and scalable video data programming. Our target users are Machine Learning (ML) practitioners dealing with video datasets that have insufficient labeled examples. They seek to supplement high-quality data samples for enhanced model performance of aimed tasks. In this paper, we mainly focus on the video classification task. We also discuss how *VideoPro* can be extended to support other tasks in Sec. 7. Drawing inspirations from the event segmentation theory [33] in cognitive science, we leverage Computer Vision (CV) techniques to decompose intricate video sequences into a series of human-comprehensible and semantically meaningful events. To address the first challenge, we propose a two-stage template mining algorithm to exploit diverse event sequential patterns as templates for labeling functions. Regarding the second challenge, the *VideoPro* interface provides carefully designed visualizations and rich interactions, allowing users to efficiently explore, validate, and refine labeling templates based on their domain knowledge. Users can then apply the labeling functions to video data at scale and make prompt adjustments during the iterative programming process. Our contributions are summarized as follows:

- We propose a novel approach that leverages advanced algorithms to exploit diverse event sequential patterns from videos to guide video data programming.
- We develop a visual analytic system that provides carefully designed visualizations and rich interactions to facilitate efficient and scalable video programming.
- We conduct two case studies and expert interviews to validate the efficiency and effectiveness of the system.

## 2 RELATED WORKS

### 2.1 Interactive Data Labeling

A surge of research has been proposed to minimize the effort and accelerate the labeling process for supervised ML. These works can be categorized into model-centered and user-centered approaches [22]. Model-centered approaches, exemplified by *Active Learning* (AL), employ various selection strategies to prioritize the labeling of the most "informative" data samples, thus reducing the burden by focusing on smaller subsets of candidate instances [6]. However, AL limits users

to labeling lengthy sequences of recommended instances solely determined by the selection algorithms, causing the final model performance to be heavily influenced by the selection strategies [5].

Visual interactive labeling is a user-centered approach that takes advantage of users' domain expertise and visual perception to guide the selection and labeling process. Various visualization techniques (*e.g.*, self-organizing maps [47], dimension reduction techniques [12, 13, 32, 42], and thumbnail visualization [34, 54]) have been employed to cluster and sort similar items for efficient labeling [77]. Recent works have incorporated more model suggestions with visualizations to further enhance labeling efficiency [11, 75, 82]. For example, VIANA [59] and AILA [16] enable efficient text document labeling by visually emphasizing important text segments recommended by ML algorithms. These mixed-initiative workflows allow users to understand and steer the models by eliciting human knowledge during the interactive labeling process [29, 31]. Notably, PEAX [36] employs the iterative labeling strategy to train classifiers for searching similar patterns in multivariate time series. Despite the advancements, these approaches still face scalability challenges due to the need for manual verification of data instances one by one. We aim to address this limitation by developing a scalable solution that enables at-scale labeling and programming of video data, facilitating efficient knowledge transfer from a small set of labeled videos to a large set of unlabeled videos.

Hoque *et al.* [26] proposed the visual concept programming for image data, which is the most relevant work to ours. The method decomposes images into human-understandable visual concepts leveraging a pre-trained vision-language model. Users can program these visual concepts to inject their knowledge at scale. However, the system primarily focuses on static spatial relationships between detected objects in images, and cannot easily generalize the resulting heuristics to temporal relationships among multiple events in videos. Furthermore, it relies solely on users to explore and define labeling functions and lacks prompt feedback on the impact of programming on the model performance. To achieve a streamlined and flexible video programming workflow, we first conceptualize videos as event sequences. Then we propose a two-stage template mining algorithm to automatically generate labeling templates to be explored, examined, and applied, such that users can inject their knowledge via video programming in a scalable and interpretable manner. Additionally, we offer an interim model evaluation to guide labeling focuses.

### 2.2 Visual Event Exploration in Videos

Depending on the varying processing and target intervals, video visual analytics aims to determine the statuses in frames, detect events from scenes, and generate models for videos [28]. Recent advances in CV techniques have empowered researchers to analyze videos at the frame level (*e.g.*, object detection and recognition) and study the detected objects' behaviors and interactions over extended intervals [2]. These behaviors and interactions are often broadly defined as "events" to describe the spatial and temporal dynamics within videos [55].

Many Visual Analytics (VA) systems have been developed to analyze events in videos. Li *et al.* [37] derived anomalous events from online exam videos to support efficient proctoring. Similarly, Tang *et al.* [61] detected fraudulent events in live-streaming videos with reference to streaming moderation policies. While anomaly detection seeks to identify one anomalous event or instance as evidence, many analytical tasks require a comprehensive and multimodal context for decision-making. As Wang *et al.* [66] and Liang *et al.* [40] summarized, data in different modalities can dominate, complement, or conflict with each other. These properties have been applied to VA systems that analyze emotion [44, 79], speech [68, 69], and body language [72, 78] in videos. These systems used multimodal and heterogeneous data sources to infer the actual states of events. However, they mainly focused on one event at a time with little consideration for their temporal order, which is crucial for contextual reasoning and gaining higher-level insights.

Parry *et al.* [50] identified three characteristics of events in videos, *i.e.*, *hierarchy*, *importance*, and *state transition*. They have inspired later research to analyze videos through the lens of event sequence understanding. EventAnchor [19] is developed based on the observation

that badminton tactics are formulated by individual strokes, which can be detected by CV algorithms. From this observation, a three-level hierarchy (*i.e.*, object, event, and context) is proposed and further generalized to sports videos as the object-event-tactic framework [15] to inform the design space of augmented sports videos. As for state transition and importance, Anchorage [71] performed event sequence analysis on customer service videos to study how different states in services affect event satisfaction ratings. However, these works still analyze one video at a time and have low scalability.

We aim at the data labeling scenarios, which extrapolate the event knowledge obtained from individual videos to a collection of videos. Over the past decade, the architecture and challenges of video labeling tools have evolved from labeling visual features [18, 29] to labeling accurate event contexts [2]. Given the complexity of temporal information, these event contexts require additional information to assist careful human labeling for reliable knowledge injection. For example, users need consistency checks when coding recorded system usage videos [7] and temporal awareness when analyzing color usage in movies [24]. Similar to these video labeling tools, our approach extracts sufficient CV-based features and supports an iterative labeling process. Furthermore, we explore the use of data programming on videos, emphasizing the events and their temporal relations to form more prominent labels. We propose using event sequences to distinguish and retrieve batches of videos with specific sequential patterns of interest.

## 3 REQUIREMENT ANALYSIS

Our goal is to develop a visual analytics system that enables efficient user knowledge integration and facilitates high-quality data label generation at scale through interactive video data programming. The initial motivation for this research originated from our collaboration with two companies, aiming to develop high-performance models for real-world applications, including the analysis of customer and student behaviors in service and educational videos. Considering the diverse and complex nature of events to analyze in these domain-specific videos, domain experts need to manually label the video dataset before model training. However, the video labeling process was time-consuming, taking several weeks even for a small-scale dataset of approximately one thousand videos, due to limited expert availability and the substantial workload involved. Therefore, finding an efficient and scalable way to transfer domain knowledge from a small labeled video dataset to a large unlabeled one for high-quality data sample supplementation has been a persistent demand.

We worked closely with five ML experts (**E1**-**E5**, five males; three researchers, and two MLOps engineers) to understand the general needs and to derive design requirements. **E1**, **E2**, and **E3** are three researchers with multiple top research publications in the areas of CV and interactive ML. **E4** and **E5** are two MLOps engineers from our collaborated company who have averaged five years of experience in developing and deploying ML models. Specifically, **E1** and **E3** are the co-authors of this paper. All experts have rich experience training and utilizing ML models for video analysis. They highlighted that despite the availability of many public video datasets, building resilient models tailored to domain-specific tasks still necessitates significant amounts of real-world labeled data. Given the shortage and acquisition difficulty of such labeled data, experts expressed a desire for a tool that supports scalable knowledge transfer and efficient video programming.

The derived four design requirements are summarized as follows:

**R1 Decompose videos with meaningful temporal event sequences** All experts acknowledged the challenging and time-consuming nature of comprehending video datasets due to their large volume and rich temporal and semantic information. They emphasized the importance of presenting videos in a way that humans can readily understand and explore. Particularly, experts mentioned that video contains much redundant and unimportant information. They often rely on key events to digest the entire video content, which also echoes prior research [33, 50] on video understanding. **E1** commented that "condensing lengthy video content into a succinct event sequence enables quick grasp of the video's essence at a glance, without the need to review the entire footage."

**R2 Summarize event temporal relationships with templates from multiple facets** Given the large set of events in the video dataset, all the experts concurred that it is crucial to summarize event temporal relationships in videos with several compact templates and identify meaningful ones that can serve as labeling functions for video programming. Specifically, a template is a sequence of events shared by several videos, which can potentially help describe the semantics of the labels and define labeling functions for video programming. In addition, the experts expressed interest in exploring the templates from multiple facets, such as data coverage and model performance, to identify meaningful ones. For example, **E1** prioritized templates that yield poor model performance, while **E2** focused on templates that encompass a larger number of unlabeled instances. **E4** showed interest in templates containing instances from a single class, indicating that "such templates may well capture class-specific characteristics."

**R3 Support efficient and scalable template-guided video data programming** The experts expected the system to support interactive validation and refinement of templates to achieve efficient and scalable video programming. They pointed out that comprehending the semantic implications of templates and verifying their correctness is crucial to ensuring high-quality labeling outcomes. Moreover, the system should allow experts to refine or manually compose templates based on their domain knowledge and new insights that emerge during the exploration process. Additionally, the system should automatically retrieve the most relevant videos for programming. This will allow users to apply selected and refined templates to program a large number of videos efficiently, as **E5** commented, "it would save much effort if we could apply the knowledge to a batch of videos simultaneously."

**R4 Reveal the effect of programming on model performance** The experts also expressed a desire to monitor model performance changes throughout the programming process. They suggested that the system should provide visualizations depicting the iterative programming process to improve controllability and transparency. They can thus gain insights into the effectiveness of selected templates and data samples as well as make corresponding adjustments in the later programming stage. For instance, **E3** said that when observing an unbalanced dataset distribution, he would consider adding more data samples from the minority classes to balance it. Based on the visualized programming process, the experts can also make informed decisions about when to retrain the model and when to stop programming.

## 4 SYSTEM & METHODS

In this section, we first provide an overview of the system framework and workflow. Then we illustrate the methods for video data processing, event extraction, and labeling template mining.

### 4.1 System Framework

Figure 2 demonstrates the overarching system framework. The input video dataset consists of a small number of videos with ground truth labels and a substantial amount of unlabeled videos. The *Event extraction* module (Fig. 2A) first abstracts the input videos as temporal sequences composed of various events (*e.g.*, wave hands) that humans can readily understand. Subsequently, in the *Template mining* module (Fig. 2B), a two-stage template mining algorithm is employed to extract diverse sequential patterns among events (*i.e.*, the order of event occurrence) from the collections of output video event sequences from the *Event extraction* module. In the first stage, the sequential pattern mining algorithm (Fig. 2B-1) extracts sequential patterns, which serve as potential labeling templates for programming. In the second stage, the MinDL algorithm (Fig. 2B-2) further distinguishes and clusters the nuanced sequence variations within a template for further examination and modification. In the *VideoPro* interface, Users begin by conducting a comprehensive exploration of the generated templates in the *Template View* from multiple perspectives, including model accuracy and data coverage (Fig. 2C-1). Following the selection of a template of interest, users can then efficiently validate and refine the template (Fig. 2C-2),
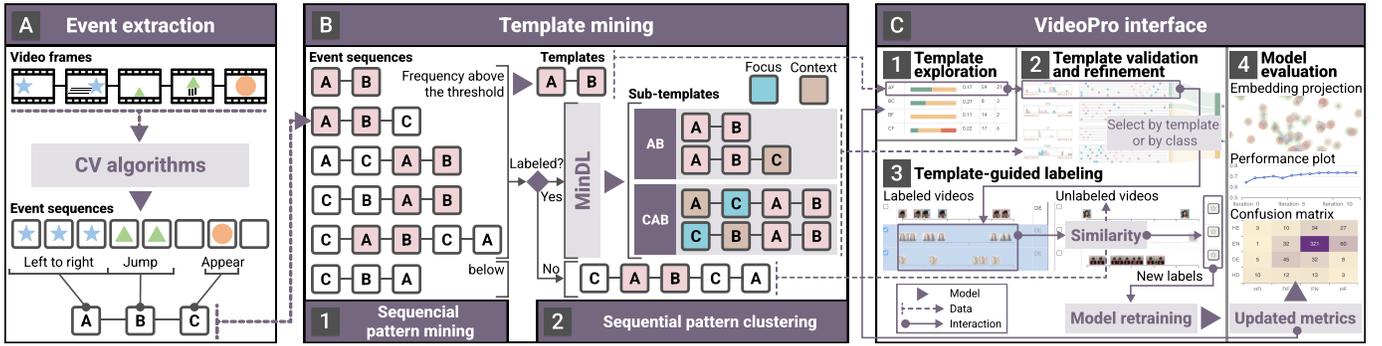
Fig. 2: The system framework contains three main modules. (A) The *Event Extraction* module converts input videos from the dataset into event sequences. (B) The *Template Mining* module distills the event sequential patterns as templates to guide programming. (C) The *VideoPro* interface supports template exploration, validation and refinement, at-scale labeling, and model evaluation for the iterative programming process.

and subsequently apply the validated and refined template to label videos at scale in the *Labeling View* (Fig. 2C-3). The labeled instances are then forwarded to the model for retraining. Users can inspect and evaluate the impact of each programming iteration on the model performance in the *Info View* (Fig. 2C-4) and correspondingly adjust their programming strategy in the subsequent iterative programming process.

## 4.2 Data and Event Extraction

Given input raw videos, state-of-the-art CV algorithms are leveraged to extract pre-defined events, which vary based on domain-specific requirements and expert needs. For instance, in application scenarios focusing on human behaviors, events of interest may include body movements (*e.g.*, jump and move right). These movements can be captured through analyzing position and angle changes of body parts based on heuristics and object detection models. Each extracted event is represented as a tuple (*eventType*, $t\_start$, $t\_end$), where *eventType* denotes the event type, and $t\_start$ and $t\_end$ are the timestamps of the start and end of the event.

## 4.3 Template Mining

Event sequential patterns, including the order and frequency of event occurrence, are crucial for comprehending and comparing video event sequences during programming. Considering the diversity and complexity of event sequential patterns, we adopted a two-stage template mining algorithm (Fig. 2B) to efficiently extract event sequential patterns and characterize the labeling templates. The two-stage template mining algorithm allows for scalable and generalizable analysis of large-scale datasets of varying lengths and diverse event sequential patterns. The first frequent sequential pattern mining algorithm [67] provides a comprehensive dataset overview and avoids generating unwieldy templates that can be challenging for experts to interpret and define. It also allows users to add self-defined constraints on template compositions flexibly to accommodate their needs [81, 83]. The MinDL algorithm [14, 70] in the later stage further summarizes and distinguishes nuanced sequence differences within a template to facilitate detailed validation and refinement.

After event extraction, each video can be construed as an event sequence, denoted as an ordered event list $S = [e_1, e_2, ....e_m]$ where $e_i$ belongs to the event set $E$. The video dataset as a whole can then be expressed as $\mathcal{S} = [S_1, S_2, ...S_n]$, where $n$ signifies the total number of video instances. A sequential pattern $P = [e_1, e_2, ...e_{|P|}]$ is a subsequence of some $S \in \mathcal{S}$ if there exist an ordered $|P|$-tuple $m = (m_1, m_2, ..., m_{|P|})$ such that $S[m_i] = e_i$ for each $e_i \in P$. For example, the sequential pattern $P = [A, D]$ is a subsequence of $S = [A, B, D, C, D]$ with two ordered 2-tuples (1,3) and (1,5). A sequential pattern is considered frequent if its occurrence exceeds a manually defined threshold. We first employed the seq2pat algorithm [67] to extract frequent sequential patterns from the video dataset $\mathcal{S}$, which were then used as labeling templates $T = [T_1, T_2, T_3...]$. This algorithm was chosen over other sequential pattern mining techniques due to its scalability and efficiency. It utilizes

a multi-valued decision diagram structure [27] to compactly encode video sequences, enabling efficient computation for large volumes of sequences (*e.g.*, thousands) in our scenario. Moreover, the algorithm is highly adaptable, allowing for flexible addition and revision of various constraints, such as sequential pattern length and continuity, based on user needs and task requirements.

We then implemented the MinDL algorithm [14, 70] to further analyze sequence nuances within a template. This algorithm applies the minimum description length principle [23] to partition video sequence collections within the selected template into clusters and summarizes each cluster with the most "representative" sequential pattern, denoted as sub-template. Events belonging to the selected template are denoted as core events. Events within the sub-template that are not part of the selected template are called focus events, while events outside the sub-template are referred to as context events (Fig. 2B-2). Every individual sequence in the cluster can be restored by editing the sub-template, including adding, deleting, or replacing events. The total description length equals the sum of the sequential pattern length and edit length, and the optimal clustering results are obtained by minimizing the total description length $L(\mathcal{C})$:

$$L(\mathcal{C}) = \sum_{(P,G)\in\mathcal{C}} \text{len}(P) + \alpha \sum_{(P,G)\in\mathcal{C}} \sum_{s\in G} \|edits(s,P)\| + \lambda \|\mathcal{C}\| \quad (1)$$

Here, $\mathcal{C}$ denotes the collection of video sequences in a template. $s$ represents the individual video event sequence. The divided sequence clusters are denoted as $\mathcal{C} = \{(P_1, G_1), (P_2, G_2), ..., (P_n, G_n)\}$ where $P_i$ and $G_i$ are the representative sequential pattern and sequence collection of the $i^{th}$ cluster. The parameters $\alpha$ and $\lambda$ respectively control the information loss importance and the number of clusters. Based on our experiment results, we found that setting $\alpha$ as 0.8 and $\lambda$ as 0 can yield a satisfactory summary for our dataset. We adopted a similar Locality Sensitive Hashing (LSH) strategy [14, 70] to speed up the computation. We also modified the original algorithm to adapt to our problem. Specifically, the computed representative sequential patterns of all clusters must include the original template for effective understanding and comparison. The MinDL algorithm excels in partitioning sequences into meaningful clusters based on temporal similarity and identifying representative sequential patterns to provide an informative summary. This is particularly useful for users to compare and understand different video sequence clusters for further labeling template validation and refinement in our scenarios.

## 5 USER INTERFACE

The *VideoPro* interface consists of three coordinated views (Fig. 1) to support flexible and smooth programming experience. In this section, we introduce the visual design of each view and the interactions connecting them in detail. The *VideoPro* adopted a unified color and event encoding scheme that is displayed at the top of the system interface. In consideration of scalability and generalizability, we use alphabets instead of icons or colors to encode individual events.

## 5.1 Template View

The *Template View* (Fig. 1A) summarizes the frequent and influential labeling templates in an organized table. It facilitates multi-faceted template exploration and comprehension (**R1, R2**).

The first column in the *Template View* records the template name, which indicates the summarized event sequential patterns. The second column uses a stacked bar chart to encode the class distribution of labeled video instances included in the corresponding template. The length of the bar chart encodes the video instance number, while the color encodes the class type. Hovering over the bars of different colors shows each class's exact number of labeled video instances, providing a clear understanding of the class distribution within the template. The bar charts will be updated after each labeling round. Newly labeled instances are visually distinguished from previously labeled ones using the corresponding class color and a check texture. The third and fourth columns respectively display the overall prediction accuracy of labeled video instances and the number of unlabeled instances within the template, which will also be updated after each labeling round.

A control panel on the top of the template table offers multiple interaction options, where users can choose to aggregate templates in different ways, including by prefix, by degree (*i.e.*, template length), and by set (*i.e.*, event collections in template). By default, templates are aggregated by prefix. Users can expand templates for further exploration by clicking the "+" symbol. Users can customize the *Template View* based on their specific needs by setting frequency and degree threshold to filter templates. They can also sort the templates by multiple predefined metrics, including overall prediction accuracy, unlabeled video instance number, and label purity in ascending or descending order. In addition, users can manually input and search for templates based on their domain knowledge in the search box above the table.

## 5.2 Labeling View

Upon selecting a template in the *Template View*, users can validate and refine the selected template, as well as examine the videos that match the template for scalable labeling in the *Labeling View* (**R1, R3**).

The upper part of the view (Fig. 1B) consists of three parts from left to right: the summary figures, the cluster heatmaps, and the connected Sankey diagrams. The summary figures (Fig. 1B-1 and Fig. 3A), inspired by the periphery plots [48], provide an overview of the temporal event distributions within the corresponding clusters. The middle stacked line charts depict the aggregated temporal distribution of the sub-template events across the entire video clusters, while the histograms on either side illustrate the frequency of context events occurring before and after the sub-template events. This design enables users to compare the event temporal distribution of sub-templates and observe the differences in contextual events between and within clusters.

The middle cluster heatmaps show the temporal distribution of the labeled videos belonging to the clusters. Each row represents an individual video sequence, and each grid represents a fixed time interval (Fig. 1B-2). For example, if one video is 10 seconds long and there are 10 grids, then each grid represents 1 second time interval. To facilitate cross-video temporal comparisons, the time duration of all video sequences is normalized so that they contain the same number of grids. Videos belonging to the same cluster are vertically stacked together, with larger clusters having larger heights. The color of each grid indicates the types of events occurring during the corresponding time interval, including core events from the selected template, the focus events in the sub-template, and other context events. Users can hover over the grid to inspect the specific event.

Furthermore, a Sankey diagram-based design (Fig. 1B-(3-4)) is adopted to visualize the label distribution across different clusters. The colored bar at the end of each video sequence indicates its label class. Therefore, the height of the colored bars at the end of each cluster (Fig. 1B-3) reflects the number of video instances belonging to the corresponding class in the cluster. The rightmost colored rectangles (Fig. 1B-4) represent corresponding classes and are linked with their contained video instances (*i.e.*, the colored bars) through flows of different widths. The width of the flows equals the bar height, thereby encoding the total number of video instances for each class. Hovering

over a rectangle will highlight all associated flows. Additionally, users can click on each rectangle to stack videos of the same class together for efficient comparison. Additionally, users can select a group of videos by clicking on the corresponding colored bar. Then the original video keyframe sequences of the selected group will be displayed below.
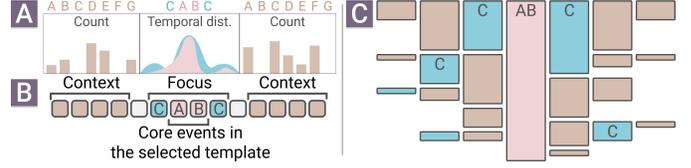


Fig. 3: The design for event sequential pattern summarization. (A) our current design based on the periphery plots [48]. (B) an illustration of the original event sequence. (C) an icicle plot alternative design.

**Alternative design** Another candidate design based on the icicle plot (Fig. 3C) was considered to visualize the sub-templates. By accumulating all event sequences in the sub-template as one, the bars' height encodes the event occurrence in order. These events are aligned by the selected template, with the before-and-after events on the sides. However, the design does not scale when the sequences are long (*i.e.*, too many layers on both sides). It could also mislead users that the left and right sub-sequences belong to some actual sequences if they share the same horizontal positions. For example, the CABC in the middle of Fig. 3C might not exist in the cluster. To fix these critical flaws, we adopt the current design that shows the overview and details separately.

The lower part of the *Labeling View* presents the original video content to facilitate quick examination and at-scale labeling (Fig. 1B-(5-6)). The lists of labeled and unlabeled videos are displayed on the left and right sides respectively, enabling straightforward comparison. Unlabeled videos are ranked based on their similarities to labeled videos by default. The similarity ($Sim_{total}$) between an unlabeled video and a labeled video is modeled by a linear combination of similarities of both event sequence ($Sim_E$) and video embedding ($Sim_V$): $Sim_{total} = w \cdot Sim_E + (1-w) \cdot Sim_V$. The $Sim_E$ is measured using editing distance to compare the discrete event sequences of the two videos, while $Sim_V$ is measured using cosine similarity to compare their video embeddings. The weight factor $w$ balances the assessment of patterns of interest ($Sim_E$) and overall video visual similarities ($Sim_V$). Users can adjust the similarity slider to control video similarity, and retrieve similar unlabeled videos by selecting corresponding labeled videos and clicking the retrieval button. In the video list, each row represents a single video. Each event within the video sequence is succinctly summarized using the extracted keyframe. The position of the keyframe on the horizontal timeline encodes the event's occurrence time, while the border color indicates the event type (core, focus, or context event). Users can hover over the keyframe to browse the complete frame sequences of the event. They can also click on the row to play the original videos for detailed inspection and bookmarking. This design allows efficient video content digestion and intuitive comparison of the event temporal distribution. Users can apply a label to multiple selected videos at once by checking corresponding selection boxes in an efficient and user-friendly manner. Users can also check the labeling history and resolve labeling conflicts in the upper-left labeling history panel.

## 5.3 Info View

The *Info View* (Fig. 1C) provides comprehensive information about data embedding distribution and model iterations (**R4**).

The *Projection* design (Fig. 1C-1) provides an overview of data instances by displaying their label status and latent space similarity. High-dimensional latent embeddings are projected onto a 2D plane using the UMAP algorithm [45], resulting in data instances with similar embeddings positioned close to each other. Labeled and unlabeled data instances are differentiated using two distinct colors. Users can select to view all data instances or focus on partial(*i.e.*, labeled or unlabeled) data instances from the top menu. A heatmap is added in the background to

encode the prediction error of data instances, with the redder shades indicating higher prediction errors.

The *Model Iteration* part (Fig. 1C-2) serves to update users about the impact of each iteration of programming on the model training progress. It includes an overall model accuracy line chart and a confusion matrix for model performance evaluation. The line chart shows how overall model accuracy changes with the number of labeled instances. The x-axis indicates the number of labeled instances while the y-axis indicates model accuracy. To ensure computation efficiency, retraining occurs when the number of newly labeled instances reaches the batch threshold, and the line chart will be updated accordingly. The confusion matrix, color-coded with a sequential colormap, shows the proportion of correctly classified video instances per class. The rows and columns represent ground truth classes and predicted classes respectively. Users can analyze classifier performance across classes, guiding template selection and data supplementation in subsequent programming iterations.

### 5.4 Cross-view Interactions

The *VideoPro* system offers diverse interactions for seamless coordination of different views with on-demand access to details.

**Clicking** Users can double-click on a specific template to inspect labeled and unlabeled video instances belonging to the template in the *Labeling View* and highlight them in the *Info View* projection plane. The reset buttons can be used to undo any operations.

**Lasso and zooming** Users can leverage the lasso and zoom interactions in the *Info View* projection to inspect and select instance groups of interest. The corresponding templates will be computed and updated in the *Template View*.

## 6 EVALUATION

In this section, we demonstrate the efficiency and effectiveness of our system through two case studies and domain expert feedback. The first case study is conducted on a real-world online education video dataset provided by our collaborated speaking training company. This dataset was used to build a robust classification model for assessing students' engagement levels in online classes, as no related public datasets or models were available. The second case study is performed on the UCF101 dataset [57], a representative public action recognition dataset, for the action classification task. The primary goal of these two case studies is to facilitate experts in efficiently supplementing high-quality data samples using *VideoPro*, achieving satisfactory model performance with minimal effort.

### 6.1 Case One: Engagement Classification

We invited expert **E1** to conduct the case study. As a member of the collaborated project, **E1** has been responsible for developing a classification model on this dataset and involved in the prototype design of our system. He thus has a good understanding of the task, dataset, workflow, and system design.

**Dataset** The whole video dataset contains 5,788 videos in total, including 1,774 videos with four-class ground-truth labels and 4,014 videos without labels. For the labeled videos, the label falls into four classes: *Highly Disengaged (HD)*, *Disengaged (DE)*, *Engaged (EN)*, and *Highly Engaged (HE)*. This classification scheme is established according to the experts' requirements and previous work practices [3, 62]. The class distribution of the labeled videos is as follows: *HD* (8.68%), *DE* (23.96%), *EN* (52.03%), and *HE* (15.33%). Following MS COCO [41], we further split the videos with ground-truth labels at the proportion around 2:1 into the training and test sets. In the splitting process, we maintain the label distribution of four classes to be the same in both the training and testing sets. The training set contains 1,182 videos, and the test set contains 592 videos.

**Initial Setting** To understand typical events for assessing student engagement levels, we interviewed three experienced teachers from our collaborating company. These teachers, with rich domain knowledge, are also responsible for labeling a small subset of the dataset. Ultimately, the consolidated event set $E$ consisted of seven types of events: active hand movement, look away, look center, smile, look

down, move away from the screen, and move close to the screen. We leveraged several state-of-the-art CV techniques [4, 8, 46] to extract these representative events from videos. Initially, we trained a baseline classifier that integrated spatiotemporal features extracted by I3D [10], a state-of-the-art pre-trained model, and event features represented by one-hot encoding. We use the accuracy for each class and the overall F1 score to evaluate the model performance. It achieved an overall F1 score of 66.78% on the test set, where its performance is recorded in the first row of Tab. 1.

**Iteration One: Distinguish between *DE* class and *EN* class** After the initial round of training, **E1** observed that the model performance was unsatisfactory in distinguishing between the *DE* class and *EN* class. He suspected that the model struggled to effectively differentiate some videos within these two classes that share similar event sequential patterns. To address this issue, **E1** aimed to identify common templates with low accuracy that were shared between the *DE* class and *EN* class.

While examining the projection in the *Info View* (Fig. 1C-1), **E1** identified a group of video embeddings highlighted with a red-colored background, indicating high errors. To further investigate these videos, **E1** utilized the lasso tool to select them, and the corresponding templates that characterized these videos were shown in the *Template View* (**R2**). **E1** observed that the template "AF" exclusively contained videos from the *DE* and *EN* classes, as indicated by the two-color distribution bar chart (Fig. 1A-1). This template also contained a relatively large number of labeled and unlabeled videos. Therefore, he decided to further investigate the "AF" template by double-clicking on it to examine its contained videos in the *Labeling View*.
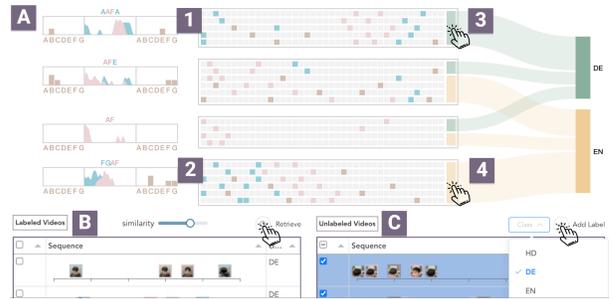


Fig. 4: The *Label View* in the case one.(A) the sub-templates within the selected "AF" template. (B) the corresponding labeled videos and retrieved unlabeled videos when clicking on the green colored bars.

Upon observing the flows between these clusters and their corresponding classes in the *Labeling View*, **E1** found that two clusters exclusively contained videos from the *DE* class (Fig. 4A-1) and *EN* class (Fig. 4A-2) respectively. **E1** also inspected the representative sequential patterns and event distributions in the left summary figure to better understand the relationship between the sequence orders within the same template and class results (**R3**). The *DE* cluster was characterized by the sequence "AAFA", while the *EN* cluster exhibited the sequential pattern "FGAF". Through analyzing the event distribution histogram, **E1** also noticed that the *DE* cluster had a higher occurrence of events involving moving far away and looking away, while the *EN* cluster had a higher occurrence of events such as looking center and moving closer to the screen.

After examining the labeled videos by clicking on the colored bars (Fig. 4A-(3-4)), **E1** observed that participants classified as *DE* frequently looked down, appeared preoccupied with their own work, and only occasionally directed their attention to the center of the screen. In contrast, participants classified as *EN* listened attentively with their eyes focused on the center of the screen most of the time, looked down for a short time, and exhibited positive behaviors like smiling. These observations led **E1** to conclude that these two summarized sequential patterns effectively characterized the *DE* and *EN* classes. Consequently, **E1** felt confident in using these two refined templates for data supplementation to highlight the differences between the *DE* and *EN* classes. By clicking on the Retrieve button (Fig. 4B), **E1** obtained the unlabeled

videos exhibiting similar patterns for efficient labeling (**R3**). Through browsing the keyframes and their border colors, **E1** quickly identified the videos that closely matched the two representative patterns (**R1**). He then selected these videos by checking the selection boxes and applying the corresponding class label to them all at once (Fig. 4C).

**E1** then initiated model retraining in the *Info View*. The results of this iteration are shown in the second row of Tab. 1. Compared with the initial baseline, the performance of the *DE* and *EN* classes improved +3.86% and +2.29% respectively. This result indicated the effective utilization of the acquired knowledge about the distinction between classes in supervising model training, which was achieved by supplementing high-quality labels using refined templates. Meanwhile, **E1** noticed that the performance of the *HD* and *HE* classes significantly dropped. Considering the absence of supervision for the other two classes in this round, he thought this outcome was reasonable. As a result, **E1** planned to augment the model's understanding of the other two classes in the next iteration (**R4**).
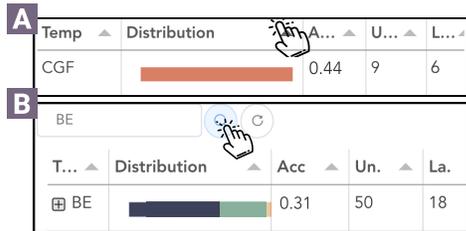


Fig. 5: The representative templates for *HE* and *HD* class. (A) the template only contains videos belonging to *HE* class. (B) the searched template based on domain knowledge.

**Iteration Two: Balance dataset distribution** After examining the labeled data distributions, **E1** noticed an imbalance across the four classes, where the *HD* and *HE* classes had only a few video samples. To improve the model's robustness and stability, **E1** decided to use *VideoPro* to supplement more samples from the two minority classes.

To identify representative templates for quick labeling of the *HE* class (**R2**), **E1** sorted the templates in the *Template View* based on the descending purity value by clicking the distribution column. The top-ranked template "CGF" displayed a red distribution bar, indicating that all labeled videos in the template belonged to the *HE* class (Fig. 5A). After randomly selecting labeled videos and browsing the original videos in the *Labeling View*, **E1** observed that this behavior sequence frequently occurred when students were deeply engaged in the teaching content. They tended to approach the screen, respond with smiles, and maintain focused attention on the screen (**R1**). As the unlabeled videos were ranked based on similarity, **E1** directly checked the last retrieved unlabeled video by hovering over its keyframes for quick validation. He found that the scenarios in this video aligned well with the labeled ones, which increased his confidence in the template. Therefore, **E1** labeled these retrieved unlabeled videos as *HE* class at scale (**R3**).

In the sorted templates based on purity value, **E1** failed to find a template with a pure blue bar in the distribution column, indicating the absence of exclusively labeled videos for the *HD* class. This observation reinforced the need to supplement more data from this class to achieve dataset balance. Drawing on his general knowledge and previous discussions with domain experts, **E1** recalled that the pattern of moving away from the screen and then consistently looking away is often associated with high disengagement. Thus he searched the corresponding template "BE" directly in the search box in the *Template View* (Fig. 5B). As a result, the template "BE" appeared at the top with a distribution bar that has a large portion of blue. Most of the retrieved unlabeled videos had a good match where he applied the *HD* label in a similar manner. After adding more samples to these two minority classes following a similar process, **E1** proceeded to send the newly supplemented samples for model retraining.

The outcomes of the second round of iteration are shown in the third row of Tab. 1. It is evident that following the introduction of

additional knowledge and supplementation of data for the two under-represented classes, the model exhibited significant improvements in its performance in these classes (+7.24% and +5.65% for *HD* class and *HE* class respectively). Moreover, the overall model performance has also been improved. After 10 programming iterations, **E1** noticed that the overall accuracy ceased to increase and instead stabilized at around 75.4%. This result also satisfied the project objective of achieving an overall classification accuracy above 70%. The final overall accuracy and each class all improved compared with the initial baseline. Consequently, **E1** is satisfied with this programming result and decided to stop programming (**R4**).

Table 1: Performance improvement using *VideoPro* on the engagement classification task, measured by the accuracy of each class and overall F1 score (larger is better).

| Results(%) Setting | HD | DE | EN | HE | F1 score |
|---|---|---|---|---|---|
| Baseline | 47.62 | 49.31 | 76.82 | 43.42 | 66.78 |
| Iteration One | 40.21 | 53.17 | 79.11 | 35.78 | 69.82 |
| Iteration Two | 47.45 | 49.04 | 78.29 | 41.43 | 70.12 |
| Iteration Ten | 55.36 | 54.07 | 80.14 | 49.64 | 75.43 |

**Post Analysis** Following the case study, a quantitative experiment was conducted to compare the labeling efficiency of *VideoPro* with an active learning-based labeling baseline approach. The baseline approach utilized the uncertainty-based strategy, a widely adopted technique in active learning [56], that selects the most uncertain videos for labeling at each time. The experiment results are summarized in Tab. 2. It showed that the active learning-based approach required labeling 2081 video samples to achieve an overall accuracy of 75.38%. In contrast, *VideoPro* enabled the expert to label 10 iterations and 452 samples in total, achieving an overall accuracy of 75.43%.

Furthermore, we compared the time cost of the two approaches. The time cost for the baseline approach was estimated based on the average time needed for labeling a single video by domain experts (i.e., teachers) using the labeling tool provided by the collaborated company. The average labeling time was half a minute per video as recorded, resulting in a total time cost of 17.3 hours. In comparison, *VideoPro* recorded the total operation time, where the expert took 1 hour to finish all the labeling. The experiment results show that *VideoPro* incurs lower labeling and time cost than the baseline approach to attain comparable levels of accuracy. It demonstrates that *VideoPro* significantly improves labeling efficiency.

Table 2: The number of labeled samples and time cost comparison between the active learning-based approach and *VideoPro*.

| Dataset | Method | # of labeled samples | time cost | F1 score(%) |
|---|---|---|---|---|
| Engagement | Baseline | 2081 | 17.3h | 75.38 |
| | VideoPro | 452 | 1.0h | 75.43 |
| UCF101 | Baseline | 496 | 0.8h | 93.92 |
| | VideoPro | 304 | 0.5h | 93.98 |

## 6.2 Case Two: Action Recognition on UCF101 dataset

To further validate the effectiveness and generalizability of *VideoPro*, we extended our system for a more general action recognition task. We invited **E6**, a sports analytics researcher who has published multiple articles about sports-related labeling and analytics tools, to conduct this case study. He has rich experience building sports analytics models and extensive knowledge in the sports and exercise domain.

**Initial Setting** For the system demonstration, the expert selected 10 sports-related action classes that he is familiar with from the UCF101 dataset. These action classes include *Archery (10.29%), CleanAndJerk*

(7.35%), *Basketball Shooting* (10.29%), *High Jump* (11.76%), *Javelin Throw* (14.71%), *Tennis Swing* (7.35%), *PullUps* (4.41%), *PushUps* (11.76%), *Lunges* (8.82%), *Body Weight Squats* (13.24%). The number in the bracket indicates the corresponding class distribution in the dataset. To identify the fine-grained semantic events associated with these activities, we conducted interviews with **E6** and his colleagues, and extensively reviewed relevant literature in the field. Drawing from experts' insights and borrowing concepts from relevant sports biomechanics research [1], we defined a set of fine-grained semantic events that include arm flexion (A), arm extension (B), arm abduction (C), arm adduction (D), leg flexion (E), leg extension (F), leg abduction (G), and leg adduction (H), body elevation (I), and body depression (J). To detect these events, we first adopted the advanced pose detection model [9] for body keypoint and part detection. We then utlized rule-based heuristics [25] to detect these events. Specifically, we calculated the displacement of body parts along and perpendicular to the body's midline to detect abduction/adduction and elevation/depression events. Additionally, we measure angle changes between body parts to detect flexion/extension events.

We followed the original training-test split of the UCF101 dataset on the selected 10 classes. The constructed 10-class dataset thus contains 1,016 videos in total, with 733 videos in the training set and 283 videos in the test set. We further split the training set into the labeled dataset with 68 videos and the unlabeled dataset with 665 videos to simulate the scenarios with very few labeled videos at the beginning. The label distribution in the original dataset is preserved during the splitting process. We adopt the state-of-the-art uniFormer backbone [39] to train a baseline classification model on the constructed labeled dataset (with 68 videos). It achieves an overall F1 score of 82.69% on the test set.

**Programming Process** After analyzing the performance of the baseline model on the test set, **E6** observed that the model performed poorly on the *High Jump* and *Javelin Throw* classes. The confusion matrix further indicated the model's inability to distinguish between these two classes. Therefore, **E6** decided to supplement more labels for these two classes. Looking at the *Template View* sorted by prediction accuracy, **E6** discovered the template "EFEFEF" (Fig. 6A), which indicates repetitive leg flexion and extension movements, contained a large portion of videos from these two classes (**R2**). Drawing from domain knowledge, **E6** pointed out the distinct stages within the *High Jump* and *Javelin Throw* activities. The *High Jump* can be roughly divided into approach, takeoff, and landing stages, while the *Javelin Throw* activity involves stages such as approach, windup, and release. These two actions shared common initial event sequences involving repetitive leg movements to generate momentum during the approach stage. Recognizing the potential value of this template in representing these two classes, **E6** proceeded to explore its contained sub-templates in the *Labeling View* by clicking on the template.
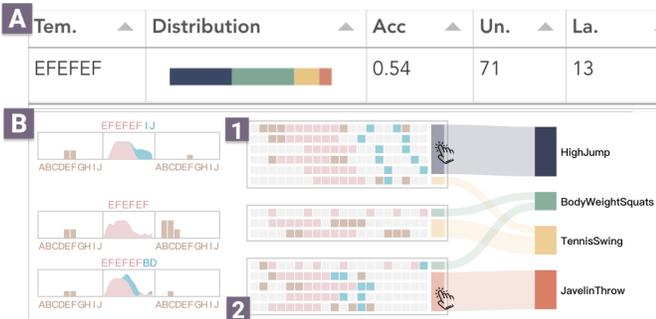
Fig. 6: (A) The template shared by *High Jump* class and *Javelin Throw* class. (B) The sub-templates in the *Labeling View* with event temporal distribution of corresponding labeled videos.

In the *Labeling View*, three sub-templates were identified. By observing the flow width and color, **E6** noticed that the sub-template "EFEFEFIJ" (Fig. 6B-1) predominantly contained videos from the *High Jump* class, while the sub-template "EFEFEFBD" (Fig. 6B-2) mainly

included videos from the *Javelin Throw* class (**R3**). This finding aligned with **E6**'s knowledge, as the event sequences following the approach stage captured the distinguishing characteristics of these two classes. The *High Jump* class exhibited a body elevation event for takeoff, followed by a body depression event for landing. On the other hand, the *Javelin Throw* action involved arm extension and adduction for the delivery and then release of the javelin.

Fig. 7: Two examples of retrieved videos for *High Jump* class and *Javelin Throw* class. (A) the video labeled as *High Jump*. (B) the video labeled as *Javelin Throw*.

**E6** then clicked on the colored bars at the end of the two clusters respectively to retrieve similar unlabeled videos for labeling from the below video list. **E6** randomly checked several videos (Fig. 7), observing the event sequence through the frame border color and hovering on the keyframes to unfold the frame sequences for quick video browsing (**R1**). After selecting the matched videos, he applied the labels to them at once (**R3**). After supplementing more samples for the *High Jump* class and *Javelin Throw* class, **E6** initiated model retraining, resulting in noticeable improvements in performance for these two classes (+1.25% for the *High Jump* class and +3.47% for the *Javelin Throw* class).

**E6** proceeded to program other classes with relatively low performance, such as *Tennis Swing* and *Lunges*. After 8 iterations with 304 videos being labeled, the model finally achieved an overall F1 score of 93.98% on the test set (**R4**).

**Post Analysis** We followed a similar practice in case one to quantitatively evaluate the labeling efficiency. On this public dataset, the active learning-based labeling approach requires labeling 496 samples to achieve an overall F1 score of 93.92%. For time cost estimation, we referred to Ma *et al.* [43], which reported an average of 45s for video-level action labels in a 60s video. Therefore, the time cost for the active learning-based approach is computed as 0.75 * (total time length of 496 labeled samples), which is 0.8h. In contrast, **E6** finished the whole programming in 0.5h. The results are listed in Tab. 2, which further validates the efficiency of *VideoPro*.

### 6.3 Expert Interviews

We further conducted semi-structured individual interviews with three ML practitioners (**P1-P3**) from the project development team, who have more than four years of experience in developing and operating ML models for video applications. While familiar with the project context, non of them had known or tried the system before the interview. The interviews began with an introduction to the research background and system designs. Then we demonstrated system workflow and usage with specific examples [74]. After the demonstration, we asked the practitioners to freely explore and try the system for programming on the real dataset, and express their thoughts, findings and suggestions in a think-aloud protocol. We also collected feedback from **E6** during the second case study. The feedback collected was categorized into the following three perspectives:

**System workflow** All participants confirmed the effectiveness of using human-understandable events to represent video data, which is "*intuitive and useful to understand video content*". They also appreciated the idea of extracting event sequential patterns as programming guide templates. **E6** commented, "*This tool is pretty helpful for labeling and analyzing sports tactics, as the event order directly determines the tactic type.*" Furthermore, the participants valued the tool's ability to efficiently search and retrieve videos from large-scale video datasets through flexible event composition and assembly. They emphasized

that this is particularly "*important and needed*" in real-world work scenarios, which allows them to retrieve video data samples at scale for model building and steering with minimal effort and cost.

**Visual designs and interactions** Overall, the practitioners reported the system is "*easy to use*" with intuitive visual designs and smooth interactions. The *Labeling View* is favored by all participants, where they can "*grasp the video content by glancing at the keyframes*". **P2** appreciated the sorting of videos based on similarity, making it easy to identify the most matched videos and apply labels in batches conveniently. The design of *Template View* is also well-received, especially for its rich interactions, enabling "*efficient template exploration based on different metrics.*" **P3** expressed a liking for the projection design in the *Info View*, with the intuitive background error heatmap and useful lasso interaction for selecting video groups of interest. Nevertheless, the participants found the *Labeling View* design somewhat complex, as it contained a lot of information, requiring some time for them to grasp.

**Suggestions for improvement P1** expressed the need to save the history of all selected templates for future reference. He also proposed that more events could be included to provide a more exhaustive summarization of the video content, while acknowledging the importance of focusing on critical ones. **P2** suggested that the system could provide real-time operation guidance and suggestions to reduce the learning curve. He also mentioned that more advanced strategies are needed to resolve label conflicts, which are currently being handled manually. **P3** recommended that the system should support adjusting more parameters such as learning rate and batch size on the interface. **E6** suggested the use of semantic meaningful icons or abbreviations to enhance the intuitive understanding of events.

## 7 DISCUSSION

During the development process of *VideoPro*, we have gained insights, identified limitations, and got inspirations for future exploration.

**Data-centric approach for video data programming with labeling templates** Data programming adopts a data-centric perspective to enhance data quality at scale, enabling model steering under the supervision of users' domain knowledge. While previous works have focused on temporal pattern labeling [36] or static spatial relationships in images [26], they fall short in handling the rich spatial and temporal semantic information present in videos. Our work overcomes these limitations by utilizing semantic-rich events to compose labeling functions. We employ compact labeling templates to summarize diverse events and their intricate temporal relationships, helping users to understand video data characteristics and identify semantic meaningful ones for labeling target data classes. This "video-event-template" abstraction process effectively elicits users' high-level domain knowledge for data labeling and model training. Currently, our templates mainly consider the semantics of event types and temporal orders. Future works can consider more complex semantics involving event characteristics like duration and object interactions. Meanwhile, when exploring different templates to distill meanings of event compositions, there often exists a trade-off between coverage and meaningfulness. Some templates may cover a large number of instances but introduce some noisy and meaningless ones, requiring greater effort for validation. On the contrary, some templates can accurately reflect the semantic meaning of a target data class but cover only a few instances. Future systems can also consider adaptive designs of templates that strike a balance between coverage and meaningfulness.

**System generalizability** The proposed generic labeling workflow is capable of accommodating various tasks beyond classification with minor modifications. For example, for temporal action localization (which seeks to identify the interval of a specific activity in untrimmed videos), *VideoPro* can match the activity with its representative event sequences to provide a rough estimate of time spans. Then, the *Labeling View* can be revised to enable zooming in on the fine-grained components of the start and end events for precise start and end timestamp annotation. For tasks such as video retrieval [76] and generation [51], the template mining algorithm and the *Template View* design can be directly employed to define and compose sequential event relationships flexibly. Moreover, considering the fundamental role of event sequences in video

data, *VideoPro* is transferable to a wide range of applications and video types. For example, domains such as social science and behavior psychology share similar requirements for building models to analyze user behaviors and interactions in recorded experiment videos. The system also readily supports needs such as sports tactics analysis, tutorial video understanding, and surgical video comparison. Furthermore, events can be compiled in different ways to create new classes flexibly for new use cases. For instance, altering the cooking order of ingredients can build templates for new recipes.

**Event extraction effectiveness** Defining atomic events properly relies on domain knowledge, task specifics, and suitable algorithms, considering the hierarchical nature of events. For instance, a cooking video for "preparing a salad" involves atomic events such as chopping vegetables, tossing, and dressing the salad. These high-level events can either be detected by action recognition [17], or further decomposed as hand movements and object manipulations that can be deduced through heuristics [30]. Visual-language models have also emerged as a powerful tool for tagging semantic concepts [21, 65] (*e.g.*, objects, actions, and scenes), offering new possibilities for capturing complex higher-level semantic events. Apart from employing more powerful models, visualizations should be designed for summarizing high-level event semantics and facilitating intuitive reviews of detailed video frames. High-level semantics representations (*e.g.*, object-scene graphs [49]) can also benefit from novel visual designs to analyze event relationships. Considering the impact of imperfect algorithms, camera movements, and view occlusions on event detection, incorporating more robust algorithms and uncertainty visualization techniques can enhance the system's resilience and reliability.

**System scalability** In terms of visual design, when the number of classes and event categories reaches tens and hundreds, it will lead to a long template list and visual clutters in the distribution bar charts in the *Template View*. To address this, *VideoPro* offers sorting and filtering options based on multiple thresholds and metrics, allowing users to quickly explore and locate templates of interest. Similarly, the Sankey diagram design in the *Labeling View* may become visually cluttered with a large number of classes. However, since experts often need to compare only a few classes at the same time, the *Labeling View* can satisfy their requirements. In the future, we plan to implement multi-level grouping strategies, together with hierarchical visualization and interaction techniques to further enhance visual scalability. For example, we can group classes and events based on some taxonomies, themes, or model performance. Then users can explore and program different video data subsets that contain a few categories of interest.

**Limitations and future works** Currently, *VideoPro* is designed for discrete events and could face challenges with datasets featuring longer, overlapping events. In such cases, events could be weighted based on their significance or aggregated into compounds (*e.g.*, $A(AB)B \rightarrow ACB$) to retain sequential patterns. However, these adjustments may complicate template mining, making additional research necessary for overlapping events [79]. Additionally, the system currently supports video programming through visual channels only, while some video analyses could benefit from incorporating concurrent audio and speech information [71]. Therefore, efficient methods of encoding and complementing multimodal information [66] for programming are worth exploring. Furthermore, the current system is designed for single-person operations. To enable collaborative programming, it is important to explore methods for efficiently resolving label conflict and maintaining consistent labeling quality in future work.

## 8 CONCLUSION

This paper presents *VideoPro*, a novel visual analytics approach that extracts and externalizes video event composition knowledge to streamline video data programming. The conducted two case studies and expert interviews validate the system's efficiency and effectiveness for video data supplementation and model steering. Meanwhile, the development and evaluation of *VideoPro* reveal several promising future research directions, including integrating more complex event attributes, balancing template coverage and meaningfulness, and exploring multimodal and collaborative video programming techniques.

## REFERENCES

[1] B. Abernethy, V. Kippers, and S. J. Hanrahan. *Biophysical foundations of human movement*. Human Kinetics, 2013. 8

[2] S. Afzal, S. Ghani, M. M. Hittawe, S. F. Rashid, O. M. Knio, M. Hadwiger, and I. Hoteit. Visualization and visual analytics approaches for image and video datasets: A survey. *ACM Transactions on Interactive Intelligent Systems*, 13(1):1–41, 2023. doi: 10.1145/3576935 2, 3

[3] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *Proc. ACL*, pp. 2236–2246. ACL, Melbourne, Australia, 2018. doi: 10.18653/v1/P18-1208 6

[4] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Openface: An open source facial behavior analysis toolkit. In *Proc. WACV*, pp. 1–10. IEEE, Los Alamitos, 2016. doi: 10.1109/WACV.2016.7477553 6

[5] J. Bernard, M. Hutter, M. Sedlmair, M. Zeppelzauer, and T. Munzner. A taxonomy of property measures to unify active learning and human-centered approaches to data labeling. *ACM Transactions on Interactive Intelligent Systems*, 11(3–4):1–42, sep 2021. doi: 10.1145/3439333 2

[6] J. Bernard, M. Zeppelzauer, M. Lehmann, M. Müller, and M. Sedlmair. Towards user-centered active learning algorithms. *Computer Graphics Forum*, 37(3):121–132, 2018. doi: 10.1111/cgf.13406 2

[7] T. Blascheck, F. Beck, S. Baltes, T. Ertl, and D. Weiskopf. Visual analysis and coding of data-rich user behavior. In *Proc. VAST*, pp. 141–150. IEEE, Los Alamitos, 2016. doi: 10.1109/VAST.2016.7883520 3

[8] Cansik. "yolo-hand-detection," github.com. https://github.com/cansik/yolo-hand-detection, 2022. (accessed Jun. 21, 2023). 6

[9] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2019. doi: 10.1109/TPAMI.2019.2929257 8

[10] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, pp. 4724–4733. IEEE, Los Alamitos, 2017. doi: 10.1109/CVPR.2017.502 6

[11] C. Chen, Z. Wang, J. Wu, X. Wang, L.-Z. Guo, Y.-F. Li, and S. Liu. Interactive graph construction for graph-based semi-supervised learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(9):3701–3716, 2021. doi: 10.1109/TVCG.2021.3084694 2

[12] C. Chen, J. Wu, X. Wang, S. Xiang, S.-H. Zhang, Q. Tang, and S. Liu. Towards better caption supervision for object detection. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1941–1954, 2022. doi: 10.1109/TVCG.2021.3138933 2

[13] C. Chen, J. Yuan, Y. Lu, Y. Liu, H. Su, S. Yuan, and S. Liu. OoDAnalyzer: Interactive analysis of out-of-distribution samples. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3335–3349, 2021. doi: 10.1109/TVCG.2020.2973258 2

[14] Y. Chen, P. Xu, and L. Ren. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):45–55, 2018. doi: 10.1109/TVCG.2017.2745083 4

[15] Z. Chen et al. Augmenting sports videos with viscommentator. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):824–834, 2022. doi: 10.1109/TVCG.2021.3114806 3

[16] M. Choi, C. Park, S. Yang, Y. Kim, J. Choo, and S. R. Hong. Aila: Attentive interactive labeling assistant for document classification through attention-based deep neural networks. In *Proc. CHI*, pp. 1–12. ACM, New York, 2019. doi: 10.1145/3290605.3300460 2

[17] M. Contributors. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020. 9

[18] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris. *A Survey of Semantic Image and Video Annotation Tools*, pp. 196–239. Springer, Cham, Switzerland, 2011. doi: 10.1007/978-3-642-20795-2_8 3

[19] D. Deng et al. Eventanchor: Reducing human interactions in event annotation of racket sports videos. In *Proc. CHI*, pp. 1–13. ACM, New York, 2021. doi: 10.1145/3411764.3445431 2

[20] L. J. et al. New generation deep learning for video object detection: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3195–3215, 2022. doi: 10.1109/TNNLS.2021.3053249 1

[21] Y. Feng, X. Wang, K. K. Wong, S. Wang, Y. Lu, M. Zhu, B. Wang, and W. Chen. Promptmagician: Interactive prompt engineering for text-to-image creation. *arXiv*, 2023. doi: 10.48550/arXiv.2307.09036 9

[22] B. Grimmeisen, M. Chegini, and A. Theissler. Visgil: machine learning-based visual guidance for interactive labeling. *The Visual Computer*, pp. 1–23, 2022. doi: 10.1007/s00371-022-02648-2 2

[23] P. D. Grünwald. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. The MIT Press, 2007. 4

[24] G. Halter, R. Ballester-Ripoll, B. Flueckiger, and R. Pajarola. Vian: A visual annotation tool for film analysis. *Computer Graphics Forum*, 38(3):119–129, 2019. doi: 10.1111/cgf.13676 3

[25] J. Hamill and K. M. Knutzen. *Biomechanical basis of human movement*. Lippincott Williams & Wilkins, 2006. 8

[26] M. N. Hoque, W. He, A. K. Shekar, L. Gou, and L. Ren. Visual concept programming: A visual analytics approach to injecting human intelligence at scale. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):74–83, 2023. doi: 10.1109/TVCG.2022.3209466 2, 9

[27] A. Hosseininasab, W.-J. van Hoeve, and A. A. Cire. Constraint-based sequential pattern mining with decision diagrams. In *Proc. AAAI*, pp. 1495–1502. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33011495 4

[28] B. Höferlin, M. Höferlin, G. Heidemann, and D. Weiskopf. Scalable video visual analytics. *Information Visualization*, 14(1):10–26, 2015. doi: 10.1177/1473871613488571 2

[29] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. Inter-active learning of ad-hoc classifiers for video visual analytics. In *Proc. VAST*, pp. 23–32. IEEE, Los Alamitos, 2012. doi: 10.1109/VAST.2012.6400492 2, 3

[30] J. Ji, R. Krishna, F.-F. Li, and J. C. Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proc. CVPR*, pp. 10236–10247. IEEE, Los Alamitos, 2020. doi: 10.1109/CVPR42600.2020.01025 9

[31] S. Jia, Z. Li, N. Chen, and J. Zhang. Towards visual explainable active learning for zero-shot classification. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):791–801, 2022. doi: 10.1109/TVCG.2021.3114793 2

[32] M. Khayat, M. Karimzadeh, J. Zhao, and D. S. Ebert. Vassl: A visual analytics toolkit for social spambot labeling. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):874–883, 2020. doi: 10.1109/TVCG.2019.2934266 2

[33] C. A. Kurby and J. M. Zacks. Segmentation in the perception and memory of events. *Trends in cognitive sciences*, 12(2):72–79, 2008. doi: 10.1016/j.tics.2007.11.004 2, 3

[34] K. Kurzhals, M. Hlawatsch, C. Seeger, and D. Weiskopf. Visual analytics for mobile eye tracking. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):301–310, 2017. doi: 10.1109/TVCG.2016.2598695 2

[35] W. S. Laseck, M. Gordon, D. Koutra, M. F. Jung, S. P. Dow, and J. P. Bigham. Glance: Rapidly coding behavioral video with the crowd. In *Proc. UIST*, pp. 551–562. ACM, New York, 2014. doi: 10.1145/2642918.2647367 1

[36] F. Lekschas, B. Peterson, D. Haehn, E. Ma, N. Gehlenborg, and H. Pfister. Peax: Interactive visual pattern search in sequential data using unsupervised deep representation learning. *Computer Graphics Forum*, 39(3):167–179, 2020. doi: 10.1111/cgf.13971 2, 9

[37] H. Li, M. Xu, Y. Wang, H. Wei, and H. Qu. A visual analytics approach to facilitate the proctoring of online exams. In *Proc. CHI*, pp. 1–17. ACM, New York, 2021. doi: 10.1145/3411764.3445294 2

[38] J. Li, H. Ding, J. Shang, J. McAuley, and Z. Feng. Weakly supervised named entity tagging with learnable logical rules. In *Proc. IJCNLP*, pp. 4568–4581. ACL, New York, 2021. doi: 10.18653/v1/2021.acl-long.352 2

[39] K. Li, Y. Wang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv*, 2022. doi: 10.48550/arXiv.2201.04676 8

[40] P. P. Liang, Y. Lyu, G. Chhablani, N. Jain, Z. Deng, X. Wang, L.-P. Morency, and R. Salakhutdinov. Multiviz: Towards visualizing and understanding multimodal models. In *International Conference on Learning Representations*, 2023. 2

[41] T.-Y. Lin et al. Microsoft coco: Common objects in context. In *Proc. ECCV*, pp. 740–755. Springer, Cham, Switzerland, 2014. doi: 10.1007/978-3-319-10602-1_48 6

[42] S. Liu, C. Chen, Y. Lu, F. Ouyang, and B. Wang. An interactive method to improve crowdsourced annotations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):235–245, 2019. doi: 10.1109/tvcg.2018. 2864843 2

[43] F. Ma et al. Sf-net: Single-frame supervision for temporal action localization. In *Proc. ECCV*, pp. 420–437. Springer, Cham, Switzerland, 2020. doi: 10.1007/978-3-030-58548-8_25 8

[44] K. Maher et al. E-ffective: A visual analytic system for exploring the emotion and effectiveness of inspirational speeches. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):508–517, 2022. doi: 10. 1109/TVCG.2021.3114789 2

[45] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv*, 2020. doi: 10. 48550/arXiv.1802.03426 5

[46] Meng1994412. "smile_detection," github.com. https://github.com/ meng1994412/Smile_Detection, 2022. (accessed Jun. 21, 2023). 6

[47] J. Moehrmann, S. Bernstein, T. Schlegel, G. Werner, and G. Heidemann. Improving the usability of hierarchical representations for interactively labeling large image data sets. In *Proc. HCII*, pp. 618–627. Springer, Cham, Switzerland, 2011. 2

[48] B. Morrow, T. Manz, A. E. Chung, N. Gehlenborg, and D. Gotz. Periphery plots for contextualizing heterogeneous time-based charts. In *IEEE Visualization Conference (VIS)*, pp. 1–5, 2019. doi: 10.1109/VISUAL.2019. 8933582 5

[49] Y. Ou, L. Mi, and Z. Chen. Object-relation reasoning graph for action recognition. In *Proc. CVPR*, pp. 20133–20142, 2022. 9

[50] M. Parry, P. Legg, D. H. Chung, I. Griffiths, and M. Chen. Hierarchical event selection for video storyboards with a case study on snooker video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1747–1756, 2011. doi: 10.1109/TVCG.2011.208 2, 3

[51] W. Price, C. Vondrick, and D. Damen. Unweavenet: Unweaving activity stories. In *Proc. CVPR*, pp. 13770–13779. IEEE, Los Alamitos, 2022. doi: 10.1109/CVPR52688.2022.01340 9

[52] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow.*, pp. 269–282, 2017. doi: 10.14778/3157794.3157797 2

[53] A. Ratner, C. D. Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Proc. NeurIPS*, pp. 3574–3582. Curran Associates Inc., Red Hook, NY, USA, 2016. 2

[54] O. Rooij, J. van Wijk, and M. Worring. Mediatable: Interactive categorization of multimedia collections. *IEEE Computer Graphics and Applications*, 30(5):42–51, 2010. doi: 10.1109/MCG.2010.66 2

[55] J. Schöning and G. Heidemann. Visual video analytics for interactive video content analysis. In *Advances in Information and Communication Networks*, pp. 346–360. Springer, Cham, Switzerland, 2019. doi: 10. 1007/978-3-030-03402-3_23 2

[56] B. Settles. Active learning literature survey. *Machine Learning*, 15(2):201– 221, 1994. 7

[57] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. doi: 10.48550/arXiv. 1212.0402 6

[58] E. J. Soure, E. Kuang, M. Fan, and J. Zhao. Coux: Collaborative visual analysis of think-aloud usability test videos for digital interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):643–653, 2022. doi: 10.1109/TVCG.2021.3114822 1

[59] F. Sperrle, R. Sevastjanova, R. Kehlbeck, and M. El-Assady. Viana: Visual interactive annotation of argumentation. In *Proc. VAST*, pp. 11–22. IEEE, Los Alamitos, 2019. doi: 10.1109/VAST47406.2019.8986917 2

[60] J. Sun, A. Kennedy, E. Zhan, D. Anderson, Y. Yue, and P. Perona. Task programming: Learning data efficient behavior representations. In *Proc. CVPR*, pp. 2875–2884. IEEE, Los Alamitos, 2021. doi: 10.1109/ CVPR46437.2021.00290 2

[61] T. Tang, Y. Wu, Y. Wu, L. Yu, and Y. Li. Videomoderator: A risk-aware framework for multimodal video moderation in e-commerce. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):846–856, 2022. doi: 10.1109/TVCG.2021.3114781 2

[62] Y.-H. H. Tsai et al. Multimodal transformer for unaligned multimodal language sequences. In *Proc. ACL*, pp. 6558–6569. ACL, Florence, Italy, 2019. doi: 10.18653/v1/P19-1656 6

[63] E. Vahdani and Y. Tian. Deep learning-based action detection in untrimmed videos: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4302–4320, 2023. doi: 10.1109/TPAMI. 2022.3193611 1

[64] J. Wang et al. Tac-valuer: Knowledge-based stroke evaluation in table tennis. In *Proc. SIGKDD*, pp. 3688–3696. ACM, New York, 2021. doi: 10.1145/3447548.3467104 1

[65] M. Wang, J. Xing, and Y. Liu. Actionclip: A new paradigm for video action recognition. *arXiv*, 2021. doi: 10.48550/arXiv.2109.08472 9

[66] X. Wang, J. He, Z. Jin, M. Yang, Y. Wang, and H. Qu. M2lens: Visualizing and explaining multimodal models for sentiment analysis. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):802–812, 2022. doi: 10.1109/TVCG.2021.3114794 2, 9

[67] X. Wang, A. Hosseininasab, P. Colunga, S. Kadıoğlu, and W.-J. van Hoeve. Seq2pat: Sequence-to-pattern generation for constraint-based sequential pattern mining. *Proc. AAAI*, 36:12665–12671, 2022. doi: 10.1609/aaai. v36i11.21542 4

[68] X. Wang, Y. Ming, T. Wu, H. Zeng, Y. Wang, and H. Qu. Dehumor: Visual analytics for decomposing humor. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4609–4623, 2022. doi: 10.1109/TVCG. 2021.3097709 2

[69] X. Wang, H. Zeng, Y. Wang, A. Wu, Z. Sun, X. Ma, and H. Qu. Voicecoach: Interactive evidence-based training for voice modulation skills in public speaking. In *Proc. CHI*, pp. 1–12. ACM, New York, 2020. doi: 10.1145/ 3313831.3376726 2

[70] Y. Wang et al. Interactive visual exploration of longitudinal historical career mobility data. *IEEE Transactions on Visualization and Computer Graphics*, 28(10):3441–3455, 2022. doi: 10.1109/TVCG.2021.3067200 4

[71] K. K. Wong, X. Wang, Y. Wang, J. He, R. Zhang, and H. Qu. Anchorage: Visual analysis of satisfaction in customer service videos via anchor events. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–13, 2023. doi: 10.1109/TVCG.2023.3245609 3, 9

[72] A. Wu and H. Qu. Multimodal analysis of video collections: Visual exploration of presentation techniques in ted talks. *IEEE Transactions on Visualization and Computer Graphics*, 26(7):2429–2442, 2020. doi: 10. 1109/TVCG.2018.2889081 2

[73] J. Wu, D. Liu, Z. Guo, and Y. Wu. Rasipam: Interactive pattern mining of multivariate event sequences in racket sports. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):940–950, 2023. doi: 10. 1109/TVCG.2022.3209452 1

[74] L. Yang, C. Xiong, J. K. Wong, A. Wu, and H. Qu. Explaining with examples: Lessons learned from crowdsourced introductory description of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 29(3):1638–1650, 2023. doi: 10.1109/TVCG.2021. 3128157 8

[75] W. Yang et al. Diagnosing ensemble few-shot classifiers. *IEEE Transactions on Visualization and Computer Graphics*, 28(9):3292–3306, 2022. doi: 10.1109/TVCG.2022.3182488 2

[76] Y. Yang, J. Kim, A. Panagopoulou, M. Yatskar, and C. Callison-Burch. Induce, edit, retrieve: Language grounded multimodal schema for instructional video retrieval. *arXiv*, 2021. doi: 10.48550/arXiv.2111.09276 9

[77] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7:3–36, 2021. doi: 10.1007/s41095-020-0191-7 2

[78] H. Zeng, X. Wang, Y. Wang, A. Wu, T.-C. Pong, and H. Qu. Gesturelens: Visual analysis of gestures in presentation videos. *IEEE Transactions on Visualization and Computer Graphics*, pp. 3685–3697, 2022. doi: 10. 1109/TVCG.2022.3169175 2

[79] H. Zeng, X. Wang, A. Wu, Y. Wang, Q. Li, A. Endert, and H. Qu. Emoco: Visual analysis of emotion coherence in presentation videos. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):927–937, 2020. doi: 10.1109/TVCG.2019.2934656 2, 9

[80] J. Zhang, C. Hsieh, Y. Yu, C. Zhang, and A. Ratner. A survey on programmatic weak supervision. *arXiv*, 2022. doi: 10.48550/ARXIV.2202.05433 2

[81] W. Zhang et al. Cohortva: A visual analytic system for interactive exploration of cohorts based on historical data. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):756–766, 2023. doi: 10.1109/TVCG. 2022.3209483 4

[82] Y. Zhang, Y. Wang, H. Zhang, B. Zhu, S. Chen, and D. Zhang. Onelabeler: A flexible system for building data labeling tools. In *Proc. CHI*, pp. 1–22. ACM, New York, 2022. doi: 10.1145/3491102.3517612 2

[83] J. Zhou et al. Dpviscreator: Incorporating pattern constraints to privacy-preserving visualizations via differential privacy. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):809–819, 2023. doi: 10. 1109/TVCG.2022.3209391 4