Building Scalable Video Understanding Benchmarks through Sports

Anonymous submission

Abstract

Existing benchmarks for evaluating long video understanding falls short on multiple aspects, either lacking in scale or quality of annotations. These limitations arise from the difficulty in collecting dense annotations for long videos (*e.g.* actions, dialogues, etc.), which are often obtained by manually labeling many frames per second. In this work, we introduce an automated **A**nnotation and Video **S**tream **A**lignment **P**ipeline (abbreviated ASAP). We demonstrate the generality of ASAP by aligning unlabeled videos of four different sports (Cricket, Football, Basketball, and American Football) with their corresponding dense annotations (*i.e.* commentary) freely available on the web. Our human studies indicate that ASAP can align videos and annotations with high fidelity, precision, and speed. We then leverage ASAP's scalability to create **LCric**, a large-scale long video understanding benchmark, with over 1000 hours of densely annotated long Cricket videos (with an average sample length of $\sim$50 mins) collected at virtually **zero annotation cost**. We benchmark and analyze state-of-the-art video understanding models on **LCric** through a large set of compositional multi-choice and regression queries. We establish a human baseline that indicates significant room for new research to explore. We will publicly release code for ASAP and LCric to enable building the next generation of long video understanding datasets and models.

1 Introduction

Humans learn and master skills (*e.g.* playing guitar) by associating and reasoning over episodic memories captured over days, months and years of failed and successful attempts. Thus, building systems capable of understanding and reasoning over very long streams of visual data has been a long-standing and crucial problem in Computer Vision.

Long-horizon **V**ideo **U**nderstanding (we refer as **LVU**) is the problem of reasoning over a long stream of video data, such as understanding the plot of a movie, or analyzing the performance of a player in a lengthy game. Progress towards LVU has been greatly limited by the lack of densely annotated data. Creating an LVU benchmark requires manually annotating videos frame-by-frame, which is incredibly tedious and hard to scale. This constraint has limited the length of existing densely-annotated video understanding benchmarks (Table 1) from a few seconds (Jang et al. 2017; Sigurdsson et al. 2016; Gupta et al. 2021; Xu et al. 2016) to a few minutes (Zeng et al. 2016; Krishna et al. 2017; Wu and Krahenbuhl 2021; Zhou, Xu, and Corso 2018; Gella, Lewis, and Rohrbach 2018; Bain et al. 2020).

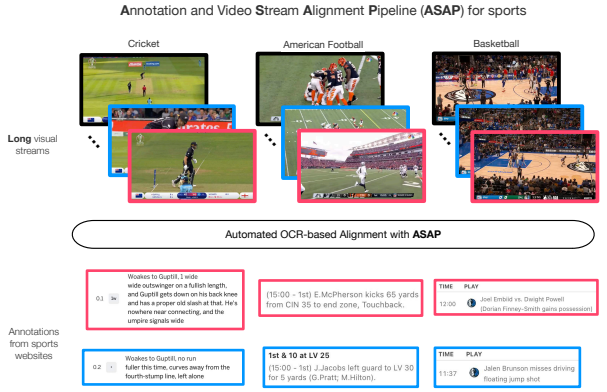

Figure 1: We propose the Annotation and Video Stream Alignment Pipeline (ASAP) for sports that uses an OCR based module to align unlabelled sports matches with structured annotations publicly available on the web. These newly aligned annotations can help compose structured queries that test for long-horizon video understanding skills.

A line of previous works (Huang et al. 2020; Tapaswi et al. 2016; Lei et al. 2018) in LVU have used readily available subtitles of TV shows or entire movies as dense annotations. While these videos are sufficiently long, manual annotations are still required to build non-trivial queries to evaluate LVU skills (Tapaswi et al. 2016; Lei et al. 2018), which greatly limits their scale. Another recent work (Wu and Krahenbuhl 2021) addresses this problem by extracting supervision from easily accessible YouTube metadata of nearly $\sim$30000 movie clips spanning $1-3$ minutes. However, the annotated clips are relatively short, and the proposed prediction tasks rely on noisy (and obscure) attributes (*e.g.* YouTube views, like-to-dislike ratio, and so on).

Sports matches are a rich source of long videos (*e.g.* a one-day Cricket match lasts nearly 8 hours) and usually have a brief scorecard embedded in the screen (as seen in Figure 1) that tracks the state of the match. Most sports matches also have dense annotations from experts available online (sports commentary describing major events in the game, *e.g.* (ESPN 2022b,a)). However, just the annotations or the

videos are not helpful individually unless they are aligned with each other.

Therefore, we introduce ASAP, an automated annotation and video stream alignment pipeline, to automatically generate video datasets with dense annotations (i.e natural language commentary, major events in the match) by aligning arbitrarily long sports matches with their commentary freely available on the web (ESPN 2022b,a)). ASAP automatically parses *match state* from the scorecard embedded in sports match videos using an OCR detector (Google 2022) and then uses the this to automatically align the videos with dense annotations available on the web. To demonstrate the generality of ASAP, we align unlabelled videos of four distinct sports (Cricket, Football/Soccer, Basketball, and American Football) with their corresponding web annotations, with an average of 95% of the annotations being aligned within ± 1 second of their occurrence in the video.

We then leverage ASAP's scalability to create **LCric**, a large-scale LVU benchmark with 1008 hours of densely annotated Cricket videos at virtually zero annotation cost, by auto-labelling 131 cricket matches of average length 7.5 hours, containing nearly 475 timestamp recordings (balls per match) on average. To our knowledge, **LCric** is the first automatically-labelled sports video dataset that contains play-by-play annotations that span entire matches. To comprehensively evaluate LVU on **LCric**, we automatically curate multiple-choice (binary and N-way) and regression queries through simple composition with boolean operations, which require varying lengths of context to answer. These queries are complex and require context aggregation ranging anywhere from 5 minutes to an hour of continuous playtime (video). In the past, such compositional query building has been leveraged in popular vision and language datasets (*e.g*. CLEVR (Johnson et al. 2017), GQA (Hudson and Manning 2019)).

We benchmark two recent state-of-the-art LVU models TQN (Zhang, Gupta, and Zisserman 2021a), MemViT (Wu et al. 2022a) on **LCric**, and find that their performance is significantly worse than our human baseline ($\sim$38% drop on query reasoning accuracy when evaluated on very long clips containing $\sim$50 minutes of playtime). This demonstrates significant room for new research to explore.

In summary, we make the following contributions – **1)** We propose ASAP, a fully automated and scalable video labeling pipeline for aligning videos of sports matches of four different sports (Cricket, Football, Basketball, and American Football) with dense annotations on the web. **2)** Using ASAP, we create **LCric**, a large-scale LVU benchmark with 1008 hours of densely annotated Cricket videos with virtually zero annotation cost. The size of **LCric** is only limited by our access to videos of Cricket matches, and we foresee the dataset being much larger given access to more videos. **3)** Finally, we benchmark the performance of two recent video understanding models on our dataset, provide ablations, and establish a human baseline on **LCric** to demonstrate significant room for modelling improvements on this benchmark. We will publicly release code for ASAP and **LCric** to enable future research on LVU.

Dataset	Avg. clip secs	# Annotations	# Hours	Autolabel
VidSitu (2021)	10s	145K	81	✗
VideoStory (2018)	18s	123K	396	✓
MSR-VTT (2016)	20s	**200K**	41	✗
Charades (2016)	30s	28K	82	✗
TGIF (2017)	30s	126K	86	✓
TVQA (2018; 2019)	75s	152K	460	✗
VTW (2016)	90s	45K	213	✓
MovieClips (2020)	120s	30K	**1270**	✓
LVU (2021)	120s	11K	**1270**	✓
YouCook II (2018)	316s	15K	176	✗
ActNet Captions (2017)	180s	100K	849	✗
LCric (ours)	**2778s**	62K	1008	✓

Table 1: Comparison among annotated datasets for benchmarking video description and video understanding methods. **LCric** has an average clip length of $\sim$2800 seconds, which is almost ten times larger than previous work, and has over 1000 hours of annotated videos.

2 Related Works

2.1 Existing benchmarks for LVU

(Wu and Krahenbuhl 2021) introduce the large-scale LVU benchmark built on movie clips and metadata publicly available on YouTube. However, the videos only range from 1-3 minutes, and the annotations are limited due to their dependence on YouTube metadata. (Oh et al. 2011) collect 29 hours of surveillance footage and bounding box annotations of major events but only have clips of length up to 3 minutes. (Corona et al. 2021) similarly collect 144 hours of surveillance footage by hiring actors to enact predefined scripts but only have clips of length up to 5 minutes. (Li et al. 2020) collect 430 videos, each 15 minutes long, and collect dense bounding box information for 80 different atomic actions. Though their videos are relatively long, our annotated videos are up to 45 minutes longer and are generated with no additional cost. (Cheng-Yang Fu and Berg 2017) collect the LoL dataset comprising 230 clips from the League of Legends video game, with each clip ranging from 30 to 50 minutes. However, they collect video highlight annotations based on very noisy and unreliable audience chat statistics. Video games also tend to have easy visual cues before major highlights that incentivize models to learn spurious correlations. Our tasks, by construction, force models to reason over a long horizon of events in a match.

2.2 Collecting dense annotations for videos

Annotating video datasets is extremely expensive. (Gupta et al. 2021; Xu et al. 2016; Sigurdsson et al. 2016) collect expensive annotations through Amazon Mechanical Turks (AMT) to label their clips with an associated text description, which greatly limits their scale (Table 1). Another line of work bootstraps from pre-existing annotations to generate new annotations. (Bain et al. 2020; Zeng et al. 2016; Gella, Lewis, and Rohrbach 2018) use pre-existing text descriptions and captions as annotations for their video clips. (Bain et al. 2020) use existing captions on YouTube and IMDb metadata to label 30000 movie clips, but assume these labels span the entirety of their clips. (Zeng et al. 2016) take user-generated titles as labels for 18100 user-generated

clips, but again assume that these labels span the entirety of their clips. (Gella, Lewis, and Rohrbach 2018) temporally align sentences from paragraph captions to social media videos to form annotated clips. Our dataset is densely annotated by temporally aligning publicly available sports annotations, which offer more structure than text descriptions and are therefore hierarchically composable, enabling the creation of queries that require large but dense context. (Liang, Xu, and Lu 2010; Liang et al. 2010; Xu et al. 2006) also align sports videos to online commentary information, however, they use heuristic methods that are not as accurate and do not scale well to generating longer and more video matches.

2.3 Video datasets based on sports

Recent interest in using computer vision to drive sports analytics (Tuyls et al. 2021) suggests the importance of a dense annotation pipeline for sports videos. Current methods for producing sports datasets involve some form of manual annotations. (Safdarnejad et al. 2015) manually label 4100 sports clips based on the given action. (Voeikov, Falaleev, and Baikulov 2020; Andriluka et al. 2017; Kazemi and Sullivan 2012) all use automatically-generated densely labelled pose annotations for sports videos but are not easily scalable because they run computationally expensive, frame-level models to generate their annotations. Larger datasets such as (Soomro, Zamir, and Shah 2012; Karpathy et al. 2014) exist but primarily focus on action recognition over a single clip, rather than a full sports video. Our dataset focuses on producing dense play-by-play annotations that span an entire sports match. Additionally, our general annotation pipeline can be easily extended to other sports, as most popular sports have rich commentary platforms on sports websites such as ESPN.

2.4 Video understanding models

Processing long videos is challenging, as it requires aggregating context over long horizons with limited computational and memory budgets. (Feichtenhofer et al. 2019) introduced SlowFast networks, which use a dual pathway operating at a low and high frame rate to enable the aggregation of context over longer horizons while capturing low-level visual attributes. (Feichtenhofer 2020) introduces a simple technique for progressive architecture expansion (along axes such as temporal, depth, width, etc.), inspired by feature selection in machine learning to achieve efficient models. Taking advantage of the implicit nature of transformers to handle long-range data (Bertasius, Wang, and Torresani 2021) proposes to adapt the standard transformer architecture for videos by enabling spatiotemporal feature learning directly from a sequence of frame-level patches. (Wu et al. 2022a) introduce MeMViT, a memory-augmented multi-scale vision Transformer, and greatly improves temporal support with minimal memory overhead and achieves state-of-the-art performance on a variety of video understanding benchmarks. While the trend shows the model's capacity to handle longer and longer video clips more efficiently, the absence of a truly long-horizon dataset inhibits a fair comparison between these baselines and also inhibits the model's transferabiity to real-world video understanding tasks.

2.5 Automated annotation pipelines

Automating annotation pipelines, even partially, is critical to developing large-scale datasets. (Ordonez, Kulkarni, and Berg 2011) introduce the SBU dataset for image-text retrieval by pruning Flickr queries to pair queries with a set of images. (Sharma et al. 2018) introduce the Conceptual Captions dataset for image captioning by leveraging the "Alt-text" HTML attribute in web images. (Desai et al. 2021) introduce the RedCaps dataset with over 12 million image-text pairs harvested from curated sub-reddits. (Pont-Tuset et al. 2020) partially automate their annotation pipeline and collect multi-modal image annotations by asking annotators to describe an image through audio while simultaneously hovering their mouse over the region they are describing. We hope that our fully automated pipeline, ASAP, will help create long and densely annotated video datasets at an unprecedented scale.

3 ASAP: Annotation and Video Stream Alignment Pipeline for Sports Matches

Sports matches provide an abundant source of long videos, with matches often lasting multiple hours. These matches also have a rich source of corresponding play-by-play annotations (i.e. expert commentary of major events in the match) easily accessible on the web (ESPN 2022b,a). These play-by-play annotations are, however, not useful standalone as they are not aligned with the video of the match. To address this, we introduce ASAP, a fully automated annotation pipeline for automatically aligning sports videos with their corresponding play-by-play annotations available on the web. ASAP automatically parses *match state* from the scorecard embedded in sports match video through an OCR detector and then uses these states to align the video with dense annotations available on the web. Thus, ASAP enables us to create long video datasets with unprecedented scale with virtually zero annotation cost and is only limited by access to sports matches. We describe ASAP in more detail below.

3.1 Stage 1: Match State Extraction

Overview. Scorecards embedded in a sport video indicates the state of the match. For instance, Basketball and Football games have a running game clock, whereas, Cricket matches have information about the *ball* being played (Section 4.1). At each video frame in the match, we wish to crop out this scorecard, and extract the *match state* contained within it using an Optical Character Recognition (OCR) system (shown in Figure 2). Once extracted, we can use this *match state* to obtain the event that took place in this frame using commentary. Next, we discuss our approach for match state extraction across all the four sports considered in our work.

Extracting Match State. In a sports video containing N video frames denoted as $[f_1, \cdots, f_N]$, we start by detecting a bounding box that encapsulates the *match state* (scorecard) information. For this, we sample a few frames uniformly

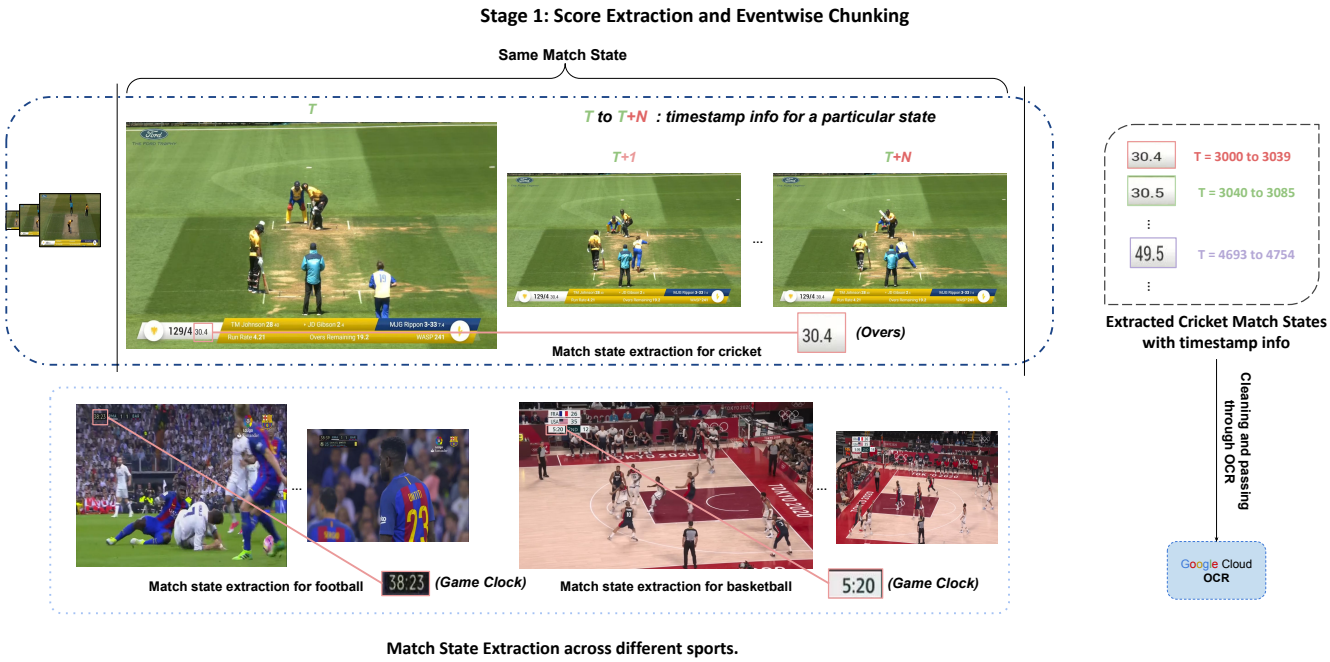

Figure 2: **Illustrating ASAP Stage 1 for Cricket matches.** ASAP automatically extracts *match state* from the scorecard of each frame with OCR, then determines what timestamps each *match state* corresponds to. Extracting such state information is often non-trivial. For instance, ASAP needs to automatically ignore frames without clear *match state* information such as the frame (marked in red) shown above due to the scorecard being occluded by a miscellaneous panel.

throughout the video and run *Google OCR* on them. Next, we determine the bounding box where text changes gradually across frames (i.e the bounding box containing scorecard). Once these bounds are detected, we crop the scorecard contained within them across all the frames. We show the crops of these scorecards across different sports in Figure 2.

Match State in Cricket (Overs). For Cricket, we represent match state by the ball that is currently being delivered. For example, in top row of Figure 2 we extract "49.4" at frame f_t, which reveals that 4th ball of 49th over is being played in this frame. Assuming this ball (event) lasted for n frames, we detect that at frame f_{t+n} the match state changes to "49.5", which means the event that took place on the 4th ball of 49th over lasted across frames $[f_t, f_{t+n}]$. In this way, we can label every single frame of the video sequence with a corresponding match state, and also club the consecutive frames across which the same match state persists.

Match State in American Football, Football, and Basketball (Timer). For remaining sports, we represent match state by the onscreen timer usually displaying minutes and seconds passed. For example, in bottom row of Figure 2, we show the time extracted across different sports. As we use videos with high frame rate of 30 FPS, we find that the same match state persists across multiple consecutive frames, and we club these frames together similar to Cricket above.

We find that locating the scorecard across frames, as well as extracting *match state* from it is non-trivial – due to the noisy and dynamic nature of the scorecard. We list some key challenges we encountered (and resolved) below.

Occlusion, and changing attributes. We found that scorecards can get occluded by advertisements, move to different position on screen, or change their attributes (format, shape, color) during gameplay. To address issues, we use a *reference scorecard*, which is a template image containing an unoccluded crop of the scorecard. We compare this reference image against all cropped scorecards in the video using a L1 distance metric, and remove spurious that lie far away from reference. These are the frames **1)** being occluded by advertisements or other miscellaneous patches, **2)** having varying attributes of the scorecard (format, shape, color, etc.). We put frames rejected by this filtering procedure in Appendix **??**. — YK

OCR calls for every frame in a video can be expensive. Sports matches last for many hours and passing all frames of a 30 FPS video through the paid *Google OCR* becomes prohibitively expensive. To reduce costs, we stack multiple scorecard boxes into a single image and get multiple annotations with a single OCR call. We also skip frames that contain the same *match state* as an older frame by detecting changes in the scorecard between consecutive frames using a $L1$ distance metric.

3.2 Stage 2: Aligning dense annotations with videos

Dense play-by-play annotations (i.e expert commentary, major events in a match etc.) are often easily available on the web (ESPN 2022b,a). Since these play-by-play annotations are indexed by the *match state* (precise *play time* or *ball* in-

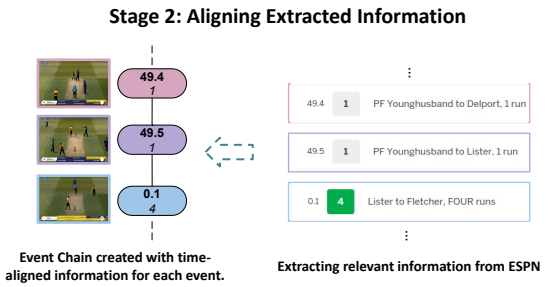

Figure 3: **Illustrating ASAP Stage 2 for Cricket matches.** ASAP scrapes dense play-by-play annotations from the web (https://www.espncricinfo.com/) and parses them into discrete events. ASAP then uses the extracted *match state* from Stage 1 to align the annotations with the video of the match.

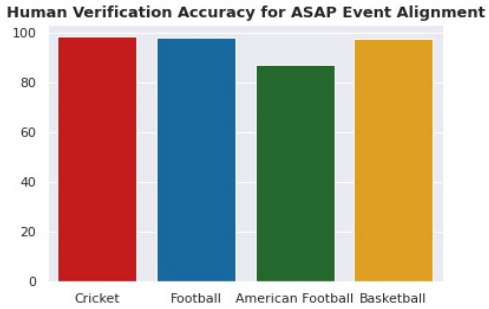

Figure 4: We find that human annotators rate annotations from ASAP to be highly accurate, with an average of 95% of the annotations being correctly aligned to the corresponding moment in a video (± 1 second) across four very different sports.

formation), we map annotations based on their *match state* to their exact timestamps (frames) found in the first stage of ASAP.

In addition to aligning the annotations with the video, ASAP also processes the sequence of play-by-play annotations into a sequence of a discrete events, which we refer to as an *event chain* (Figure 3). While some sports (Cricket and Football) already contains discrete events (e.g. 'foul', 'wicket', 'boundary' etc.) in their annotations, for other sports (American Football and Basketball), we use string-matching to parse the commentary and assign each play to a fixed event that we define (e.g. 'incomplete pass'). These extracted event chains can then be used as ground truth for evaluating LVU models. Models can be queried on different segments of the event chain of varying lengths – to test both short and long horizon reasoning. We discuss the use of event chains for evaluation in Section 4.3.

How fast is ASAP? ASAP generates annotations with very high speed and requires just 10 minutes to align and process around 7 hours of video at 30FPS on a single machine. While the optimizations we introduce in Section 3.2 greatly improve annotation speed, methods such as multiprocessing, superior similarity matching metrics for filtering frames, and OCR predictions with a locally served OCR model could greatly increase annotation speed.

How accurate is ASAP? To verify ASAP's ability to align dense annotations on the web with videos of sports matches, we conduct a study with human annotators on Amazon Mechanical Turk (AMT). We randomly sample clips from sports matches corresponding to 6 contiguous events in the event chain generated by ASAP. For all the generated clips, we ask human annotators to provide timestamps for all 6 events and then check whether the provided timestamps belong to the intervals generated for those events by ASAP. We plot the resulting accuracy of the timestamps in Figure 4 and find that ASAP is highly accurate, with an average accuracy of 95.3% across four very different sports, each differing in visual attributes, number of events and length of plays. The drop in accuracy for American football annotations can be explained by the inconsistency in the timestamps provided by ESPN. For regular plays, the timestamp indicates when the play begins; however, for 'touchdowns', the timestamp indicates when the team scored and not when the play begins. Additionally, penalties may affect the game clock, which we use to align our annotations, which sometimes leads to slight alignment issues for ASAP. We present more details in Appendix A.

4 Generating LCric dataset with ASAP

In this section, we describe how we leverage ASAP to build a long video understanding (LVU) dataset from Cricket videos online. We first provide a brief overview of the sport and describe the corresponding web annotations used by ASAP. Next, we describe a scalable approach for generating structured and compositional queries to evaluate LVU. Finally, we discuss important statistics of the queries and cricket videos in **LCric**.

4.1 Introduction to Cricket

We present a brief introduction to the sport of Cricket before introducing **LCric**. Cricket is played by two teams of 11 players each that alternate between *batting* and *fielding* throughout the game. The batting team aims to score **runs** by hitting a ball bowled by fielding team out of the playing field. Meanwhile, the fielding team aims to prevent the batting team from scoring runs and dismiss all players in the batting team by taking their *wickets*. Each exchange where the fielding team bowls a valid ball and the batting team attempts to hit the ball to score runs is called a *ball* (or *delivery*) and a sequence of 6 *balls* is called an *over*. Each *ball* is an **atomic event** and there are 12 distinct possible events per ball, listed below:

- The batting team scores n runs, where $n \in \{0, ..., 9\}$
- A wicket is taken and the current batsman is dismissed.
- A wide (invalid) ball is bowled, giving the batting team an extra run and another ball.

We present a detailed discussion of different phases of Cricket in Appendix B.

4.2 LCric: Overview

Leveraging ASAP's scalability, we create **LCric**, a large-scale LVU benchmark with 1008 hours of densely annotated Cricket videos with virtually zero annotation cost, by auto-labelling 131 cricket matches of average length $\sim$7.5 hours, containing nearly 475 timestamp recordings (balls per match) on average. ASAP automatically labels all the balls in a Cricket match with 1 of 12 events (Section 4.1) to generate a sequence of events (i.e *event chain* shown in Figure 3) for a cricket match. We then generate annotated video clips by segmenting the videos along with the aligned event chain into a contiguous sequence of 10-over ($\sim$50 minutes) clips.

4.3 LCric: Evaluating LVU with compositional queries

Motivation: An LVU system needs to possess two types of skills: a) the ability to reliably detect local (short-term) events – *e.g.*, classifying an atomic event in Cricket (say wide, wicket, or run), and b) the ability to aggregate information across these local events given a task (which we refer to as a query) – *e.g.*, counting the total number of runs scored by the batting team in an arbitrarily long video. To test for these LVU skills using **LCric**, we automatically compose and filter binary and multi-choice queries, which are then paired with long video segments. For completeness, we also experiment with a regression query.

Min-Max occurrence query. This query helps to check occurence of a single event, we first sample an atomic event from the set of all possible events, and then sample two numbers, o_{min} and o_{max}, to denote the minimum and maximum number of occurrences needed for this query to be *true*. An example of such a query is – for a given video, did a wide ball (an event) occur between 3 and 5 times inclusive?

Binary queries by chaining occurrence queries. To increase query diversity and complexity, we sample n_{chain} different min-max occurrence queries and combine them using [and]/[or] operators. For example, for a given video spanning 10-overs ($\sim$50 mins), "did a wide ball occur between 3 to 5 times [and] did a ball with 2 runs scored occur 1 to 3 times?". All **binary** queries in **LCric** are formed by chaining 1-5 different min-max occurrence queries.

Multi-choice queries by counting occurrences. We expand upon the binary occurrence queries by generating multi-choice occurrence queries, which ask models to directly predict the number of occurrences rather than predicting membership in a range. An example of such a query is – for a given video, how many times a did a wide ball occur after a ball with 4 runs? Please note that these events are sequential, but not necessarily contiguous. As most non-trivial **multi-choice** events in **LCric** occur between 0-9 times in a given clip, we use $\{0, ..., 9\}$ as our answer choices.

Filtering unbalanced queries. We can compose many LVU multi-choice and binary queries using the above formulation, however, not all queries are necessarily balanced. Due to the rarity of certain events occurring in Cricket, some queries are far easier to guess correctly than others. For example, in a 45 minutes clip (spanning 10 overs), the query – "did a ball with 2 runs occur between 0 to 10 times" is true with a probability of 87%. We filter such queries based on the probability of their occurrence in training matches, and ensure the average probability of occurrence of the selected queries to be between $0.45 - 0.55$ to avoid bias.

Regression query for counting runs. Lastly, we also experiment with a single regression query that asks the model to predict the number of runs scored as a regression output for a given video sequence.

4.4 LCric: Statistics and Dataset Splits

Statistics The dataset currently includes 1008 hours of cricket match videos across 131 unique matches (average length of 7.5 hours), along with 61957 ball-by-ball annotations. All the videos are preprocessed with a resolution of 360p and we will provide links to the source videos, which are of a higher resolution.

Dataset splits To effectively test generalization, we split all the matches in **LCric** into train, validation and test splits and ensure a '60:20:20' ratio of the number of hours in each split. Due to a limited computational budget, we present ablations on a subset of **LCric** and refer to this as **LCric-Mini**, which has around 420 hours of labelled Cricket matches, and therefore enables us to train ablations experiments in a shorter duration (2-3 days per experiment). We generate splits for **LCric-Mini** identically to **LCric**.

5 Experiments on LCric

In this section, we describe our experiments benchmarking state-of-the-art LVU models on **LCric**. We also provide a human baseline on **LCric** and demonstrate significant room for improvement. Finally, we analyze the performance of our baselines and present key insights to spur future modelling improvements on **LCric**.

5.1 Experimental Setup

Preprocessing: Following past work in LVU ((Zhang, Gupta, and Zisserman 2021b)), we process (sample) videos with a lower FPS to make training over long videos feasible. We process our longest clips (containing 10 overs of the match) at 0.1 FPS, and process clips of 2-8 overs at 0.5 FPS respectively. We **remove the scorecard** from all frames to prevent annotation leakage and process frames at a resolution of 128 x 128.

Evaluation Metrics: We compute the following evaluation metrics **1)** Classification accuracy for binary and multi-choice queries **2)** Average L1 norm for regression queries.

5.2 Baseline Models and Training Scheme

Previous works (Fan et al. 2021; Feichtenhofer et al. 2019; Feichtenhofer 2020) in LVU use pretrained CNNs (Bengio and Lecun 1997) and Transformers (Vaswani et al. 2017) paired with explicit memory modules for modelling long contexts. However, none of these methods can scale to video clips longer than a few minutes. Since our query set requires reasoning over contexts ranging up to an hour, we choose two recent state-of-the-art video understanding models, Temporal Query Network (TQN) (Zhang, Gupta, and

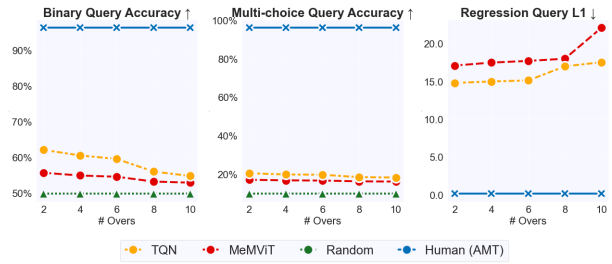

Figure 5: Accuracy (and avg. norm for regression) of various baselines on **LCric-Mini**, evaluated with our filtered set of binary, multi-choice, and regression occurrence queries for clips of over-length $\{2, 4, 6, 8, 10\}$. We find that our baseline LVU models severely degrade in performance with longer time horizons. Additionally, human performance far exceeds that of current models and therefore suggests significant room for modelling improvements on **LCric**.

Zisserman 2021b) and MeMViT (Wu et al. 2022b) as benchmarks – given their effective caching mechanisms for processing very long videos.

Temporal Query Network (TQN) (Zhang, Gupta, and Zisserman 2021b) TQN uses stochastic memory banks to efficiently model long horizon videos. They also introduce a transformer-based multi-query head to generate responses for multiple queries with a single pass through the network. TQN uses S3D (Xie et al. 2018) for the visual backbone network and processes non-overlapping contiguous sequence of 8 frames with a temporal stride of 1. Please note that the term *query* used in TQN is a modeling component and is not the same as our use of query in Section 4.

Memory-augmented Multiscale Vision Transformer (MeMViT) (Wu et al. 2022b) MeMViT applies a memory caching strategy by processing videos in an online fashion, allowing the model to efficiently store context to reason over a long horizon. MeMViT builds upon ViT (Dosovitskiy et al. 2020) by using a novel pooling method and a dynamic patch resolution approach to reduce computational costs while processing long clips. We adapt MeMViT to handle our multi-query setting by leveraging the multi-query head introduced in TQN.

Training Scheme We employ two different training schemes – **1) Homogeneous training** where we train different models for the three different types of queries (binary, multi-choice and regression) and **2) Mixed training** where we train a single model for all three types of queries.

5.3 Human Baseline

To quantify the room for modelling improvements on **LCric**, we benchmark the performance of human annotators through the AMT platform. We provide annotators with video clips from **LCric** and ask them to predict the sequence of ball-by-ball events (4.1). To compute human performance on our queries, we assume that given an event chain, humans can answer these queries by applying logical operators without mistakes.

Model	Training scheme	BC Acc. ↑	MC Acc. ↑	R L1-Norm ↓
TQN	Mixed	57.68%	19.05%	17.21
MeMViT	Mixed	54.31%	16.71%	21.79
TQN	Homogeneous	**60.74%**	**20.19%**	**10.63**
MeMViT	Homogeneous	56.53 %	17.79%	11.95
Human Perf.	--	96.34 %	96.29%	0.215

Table 2: Baseline experiment results on the full **LCric** split with **10-over clips**. BC: Binary Choice Queries, MC: Multiple-Choice Query (10-choice), R L1-Norm: Avg. L1-Norm on Regression Queries. We find that TQN performs much better than MeMViT in different training schemes across different query types. We also note that training models under the Homogeneous training scheme greatly improves performance, especially for the regression query.

Model	Segmentation Scheme	BC Acc. ↑	MC Acc. ↑	R L1-Norm ↓
TQN	Ground Truth	**80.34%**	**36.32%**	**6.89**
TQN	Uniform	60.29%	15.18%	19.31
Human Perf.	--	96.34 %	96.29%	0.215

Table 3: To understand the impact of context aggregation on task performance, we first extract event chains from 10-over clips on **LCric-Mini** by running an event classifier on different segments of the clip. We then evaluate our queries against the generated event chains and find that even with access to ground truth event segments, performance is much worse than the human baseline. BC: Binary Choice Query Set (32 queries), MC: Multiple-Choice Query (10 queries), R L1-Norm: Avg. L1 Norm on Regression Queries.

5.4 Key Results

Performance of both TQN and MeMViT degrades rapidly and approaches random for very long clips To understand the impact of length of the videos on task performance, we train different baseline models for clips with over-lengths ranging from 2 overs (∼10 minutes) to 10 overs (∼50 minutes). Figure 5 shows that performance rapidly decreases with increasing clip length and approaches the random baseline for binary and multi-choice queries. This result, in addition to the strong human baseline, demonstrates significant room for modelling improvements.

Models need to effectively aggregate context to perform well on LCric To understand the importance of aggregating context for **LCric**, we first train a TQN 'event classifier' model to predict 1 of 12 events (Section 4.1) in a video clip. The model is trained on ground truth annotations from **LCric-Mini** and has a fairly high test accuracy of 84.79%. We then divide the clips into a contiguous sequence of 'event segments' by either using ground truth segmentations from **LCric-Mini** (labeled 'Ground Truth' in Table 3) or by uniformly dividing the clips into 60 contiguous segments (labeled 'Uniform' in Table 3), as each 10-over clip contains 60 events. Finally, we leverage the learned 'event classifier' model to generate event chains by sequentially predicting events on the 'event segments' and evaluate different queries on these event chains. We report performance in Table 3 and make two observations – **1)** Although access to ground truth segments of events leads to better performance, as it aids the event classifier in making more accurate predictions, the

performance is still much worse than the human baseline. **2)** TQN model in Table 2, which aggregates context across 10-overs, performs much better than 'Uniform' in Table 3. Both these observations suggest that improving context aggregation is an important direction towards improving performance on **LCric**.

TQN performs better than MeMViT on all query types Table 2 shows that TQN performs much better than MeMViT across different query types and training schemes ($+2\%$ on binary and multi-choice queries). While we expected MeMViT to perform better due to the clever caching mechanism in its architecture, we think that adapting MeMViT with the TQN multi-query head could have caused the performance drop. We hope future work can better integrate the clever caching mechanism of MeMViT to the multi-query setting as it is infeasible to train one model for every new query.

Homogeneous training scheme leads to superior performance than Mixed training Our results in Table 2 indicate that training different models for different query types leads to much better performance than training a single model for all queries. We also observe that performance on the regression query improves by $\sim$7 points and we posit that the model needs extra representational capacity to answer the regression queries with high precision.

6 Discussion

In this work, we introduce ASAP, a fully automated annotation and video stream alignment pipeline for sports matches. ASAP automatically aligns unlabelled videos of sports matches with corresponding dense annotations (*i.e.* commentary) freely available on the web. We demonstrate the generality of ASAP by aligning unlabelled matches of four very different sports (Cricket, Football/Soccer, Basketball, and American Football) with their corresponding annotations on the web. ASAP is highly accurate across the four sports (as judged by human annotators), and is therefore robust to varying visual attributes, number of events and length of plays. We then demonstrate ASAP's potential to generate large-scale video datasets with **no additional annotation cost** by generating **LCric**, a large-scale long video understanding benchmark, with over 1000 hours of densely annotated long Cricket videos (with an average sample length of $\sim$50 minutes). We extensively benchmark state-of-the-art LVU models and establish a human baseline on **LCric**. Our strong human baseline, coupled with poor performance of state-of-the-art models, validates **LCric** as an effective benchmark for the next generation of LVU models.

Our analysis suggests that effectively integrating clever memory/caching mechanisms in our multi-query setting could lead to much better performance on **LCric**. We also find that recent LVU models struggle to reason over very long contexts and we hope the next generation of architectures can accurately localize important events and effectively reason over a very long horizon. While we demonstrate ASAP's potential by creating **LCric**, we hope future work extends, improves and leverages the various modules in ASAP to generate annotated video datasets for other sports and domains with an unprecedented scale and cost efficiency. We also hope future work can build upon our automated approach for evaluating LVU models on **LCric** and present alternate automated evaluation strategies for LVU.

References

Andriluka, M.; Iqbal, U.; Milan, A.; Insafutdinov, E.; Pishchulin, L.; Gall, J.; and Schiele, B. 2017. PoseTrack: A Benchmark for Human Pose Estimation and Tracking. *CoRR*, abs/1710.10000.

Bain, M.; Nagrani, A.; Brown, A.; and Zisserman, A. 2020. Condensed Movies: Story Based Retrieval with Contextual Embeddings. arXiv:2005.04208.

Bengio, Y.; and Lecun, Y. 1997. Convolutional Networks for Images, Speech, and Time-Series.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is Space-Time Attention All You Need for Video Understanding? In *International Conference on Machine Learning*, 813–824. PMLR.

Cheng-Yang Fu, M. B., Joon Lee; and Berg, A. C. 2017. Video Highlight Prediction Using Audience Chat Reactions. In *EMNLP*.

Corona, K.; Osterdahl, K.; Collins, R.; and Hoogs, A. 2021. MEVA: A Large-Scale Multiview, Multimodal Video Dataset for Activity Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1060–1068.

Desai, K.; Kaul, G.; Aysola, Z.; and Johnson, J. 2021. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*.

ESPN. 2022a. ESPN Soccer Commentary. https://www.espn.in/football/commentary.

ESPN. 2022b. ESPNCricinfo. www.espncricinfo.com/.

Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; and Feichtenhofer, C. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6824–6835.

Feichtenhofer, C. 2020. X3D: Expanding Architectures for Efficient Video Recognition. In *Proc. CVPR*.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6202–6211.

Gella, S.; Lewis, M.; and Rohrbach, M. 2018. A Dataset for Telling the Stories of Social Media Videos. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 968–974.

Google. 2022. Google Cloud Optical Character Recognition. https://cloud.google.com/vision/docs/ocr.

Gupta, A.; Yatskar, M.; Nevatia, R.; and Kembhavi, A. 2021. Visual Semantic Role Labeling for Video Understanding. In *CVPR 2021*.

Huang, Q.; Xiong, Y.; Rao, A.; Wang, J.; and Lin, D. 2020. MovieNet: A Holistic Dataset for Movie Understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Hudson, D. A.; and Manning, C. D. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709.

Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proc. CVPR*.

Johnson, J.; Hariharan, B.; Van Der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2901–2910.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732.

Kazemi, V.; and Sullivan, J. 2012. Using Richer Models for Articulated Pose Estimation of Footballers. In *BMVC*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-Captioning Events in Videos. In *International Conference on Computer Vision (ICCV)*.

Lei, J.; Yu, L.; Bansal, M.; and Berg, T. L. 2018. TVQA: Localized, Compositional Video Question Answering. In *EMNLP*.

Lei, J.; Yu, L.; Berg, T. L.; and Bansal, M. 2019. TVQA+: Spatio-Temporal Grounding for Video Question Answering. In *Tech Report, arXiv*.

Li, A.; Thotakuri, M.; Ross, D. A.; Carreira, J.; Vostrikov, A.; and Zisserman, A. 2020. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*.

Liang, C.; Jiang, Y.; Cheng, J.; Xu, C.; Luo, X.; Wang, J.; Fu, Y.; Lu, H.; and Ma, J. 2010. Personalized Sports Video Customization for Mobile Devices. In *Proceeding of International Conference on Multimedia Modeling (MMM)*, 614–625.

Liang, C.; Xu, C.; and Lu, H. 2010. Personalized Sports Video Customization Using Content and Context Analysis. In *International Journal of Digital Multimedia Broadcasting (IJDMB)*.

Oh, S.; Hoogs, A.; Perera, A.; Cuntoor, N.; Chen, C.-C.; Lee, J. T.; Mukherjee, S.; Aggarwal, J. K.; Lee, H.; Davis, L.; Swears, E.; Wang, X.; Ji, Q.; Reddy, K.; Shah, M.; Vondrick, C.; Pirsiavash, H.; Ramanan, D.; Yuen, J.; Torralba, A.; Song, B.; Fong, A.; Roy-Chowdhury, A.; and Desai, M. 2011. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, 3153–3160.

Ordonez, V.; Kulkarni, G.; and Berg, T. 2011. Im2Text: Describing Images Using 1 Million Captioned Photographs. In Shawe-Taylor, J.; Zemel, R.; Bartlett, P.; Pereira, F.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Pont-Tuset, J.; Uijlings, J.; Changpinyo, S.; Soricut, R.; and Ferrari, V. 2020. Connecting Vision and Language with Localized Narratives. In *ECCV*.

Safdarnejad, S. M.; Liu, X.; Udpa, L.; Andrus, B.; Wood, J.; and Craven, D. 2015. Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis. In *Proc. International Conference on Automatic Face and Gesture Recognition*. Ljubljana, Slovenia.

Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), 2556–2565. Melbourne, Australia: Association for Computational Linguistics.

Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proc. ECCV*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR*, abs/1212.0402.

Tapaswi, M.; Zhu, Y.; Stiefelhagen, R.; Torralba, A.; Urtasun, R.; and Fidler, S. 2016. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tuyls, K.; Omidshafiei, S.; Muller, P.; Wang, Z.; Connor, J.; Hennes, D.; Graham, I.; Spearman, W.; Waskett, T.; Steel, D.; Luc, P.; Recasens, A.; Galashov, A.; Thornton, G.; Elie, R.; Sprechmann, P.; Moreno, P.; Cao, K.; Garnelo, M.; Dutta, P.; Valko, M.; Heess, N.; Bridgland, A.; Pérolat, J.; De Vylder, B.; Eslami, S. M. A.; Rowland, M.; Jaegle, A.; Munos, R.; Back, T.; Ahamed, R.; Bouton, S.; Beauguerlange, N.; Broshear, J.; Graepel, T.; and Hassabis, D. 2021. Game Plan: What AI can do for Football, and What Football can do for AI.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proc. NeurIPS*.

Voeikov, R.; Falaleev, N.; and Baikulov, R. 2020. TTNet: Real-time temporal and spatial video analysis of table tennis. *CoRR*, abs/2004.09927.

Wu, C.-Y.; and Krahenbuhl, P. 2021. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1884–1894.

Wu, C.-Y.; Li, Y.; Mangalam, K.; Fan, H.; Xiong, B.; Malik, J.; and Feichtenhofer, C. 2022a. MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition. *arXiv preprint arXiv:2201.08383*.

Wu, C.-Y.; Li, Y.; Mangalam, K.; Fan, H.; Xiong, B.; Malik, J.; and Feichtenhofer, C. 2022b. MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition. *arXiv preprint arXiv:2201.08383*.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proc. ECCV*.

Xu, C.; Wang, J.; Wan, K.; Li, Y.; and Duan, L. 2006. Live sports event detection based on broadcast video and web-casting text. In *Proceeding of ACM International Conference on Multimedia*, 221–230.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5288–5296.

Zeng, K.-H.; Chen, T.-H.; Niebles, J. C.; and Sun, M. 2016. Title Generation for User Generated Videos. volume 9906. ISBN 978-3-319-46474-9.

Zhang, C.; Gupta, A.; and Zisserman, A. 2021a. Temporal Query Networks for Fine-grained Video Understanding. *arXiv preprint*.

Zhang, C.; Gupta, A.; and Zisserman, A. 2021b. Temporal query networks for fine-grained video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4486–4496.

Zhou, L.; Xu, C.; and Corso, J. J. 2018. Towards Automatic Learning of Procedures From Web Instructional Videos. In *AAAI Conference on Artificial Intelligence*.