Hierarchical Video-Moment Retrieval and Step-Captioning

Abhay Zala[*1] Jaemin Cho[*1] Satwik Kottur[2] Xilun Chen[2]
Barlas Oguz[2] Yashar Mehdad[2] Mohit Bansal[1]
UNC Chapel Hill[1] Meta AI[2]
{jmincho, aszala, mbansal}@cs.unc.edu {skottur, xilun, barlaso, mehdad}@fb.com

Abstract

There is growing interest in searching for information from large video corpora. Prior works have studied relevant tasks, such as text-based video retrieval, moment retrieval, video summarization, and video captioning in isolation, without an end-to-end setup that can jointly search from video corpora and generate summaries. Such an end-to-end setup would allow for many interesting applications, e.g.*, a text-based search that finds a relevant video from a video corpus, extracts the most relevant moment from that video, and segments the moment into important steps with captions. To address this, we present the* HiREST *(**HI**erarchical **RE**trieval and **ST**ep-captioning) dataset and propose a new benchmark that covers hierarchical information retrieval and visual/textual stepwise summarization from an instructional video corpus.* HiREST *consists of 3.4K text-video pairs from an instructional video dataset, where 1.1K videos have annotations of moment spans relevant to text query and breakdown of each moment into key instruction steps with caption and timestamps (totaling 8.6K step captions). Our hierarchical benchmark consists of video retrieval, moment retrieval, and two novel moment segmentation and step captioning tasks. In moment segmentation, models break down a video moment into instruction steps and identify start-end boundaries. In step captioning, models generate a textual summary for each step. We also present starting point task-specific and end-to-end joint baseline models for our new benchmark. While the baseline models show some promising results, there still exists large room for future improvement by the community.*[1]

1. Introduction

Encouraged by the easy access to smartphones, recording software, and video hosting platforms, people are increasingly accumulating videos of all kinds. To fuel the subsequent growing interest in using machine learning systems to extract and summarize important information from these large video corpora based on text queries, progress has been made in video retrieval [2, 17, 18, 41, 42], moment retrieval [10, 16, 17], video summarization [9, 24, 33, 34], and video captioning [13, 20, 41, 42]. Previous works have generally focused on solving these tasks independently; however, all these tasks share the common goal of retrieving information from a video corpus, at different levels of scales and via different modalities. Hence, in this work, we introduce a new hierarchical benchmark that combines all four tasks to enable novel and useful real-world applications. For example, a text-based search service that finds a relevant video from a large video corpus, extracts the most relevant moment from that video, segments the moment into important steps, and captions them for easy indexing and retrieval. To support this, we introduce HiREST, a hierarchical instructional video dataset for a holistic benchmark of information retrieval from a video corpus (see Sec. 3). HiREST consists of four annotations: 1) $3.4K$ pairs of text query about open-domain instructions (*e.g.*, *'how to make glow in the dark slime'*) and videos, 2) relevant moment timestamps inside the $1.1K$ videos, where only a part of the video ($< 75\%$) is relevant to the text query, 3) moment breakdown in several instructional steps with timestamps (7.6 steps per video, total $8.6K$ steps), and, 4) an manually curated English caption for each step (*e.g. 'pour shampoo in container'*). We collect fine-grained step-wise annotations of HiREST in a two-step annotation process with online crowdworkers on instructional text-video pairs from the HowTo100M [23] dataset (see Sec. 3.1). The instructional videos often come with clear step-by-step instructions, allowing fine-grained segmentation of the videos into short steps. While there are existing video datasets with step annotations, they are based on a small number of predefined task names [36, 46] (thus step captions are not diverse), or are limited to a single topic (*e.g.* cooking [45]). HiREST covers various domains and provides diverse step captions with timestamps written by human annotators (see Table 1), presenting new challenging and realistic benchmarks for hi-

[*] equal contribution
[1] code and data: https://github.com/j-min/HiREST

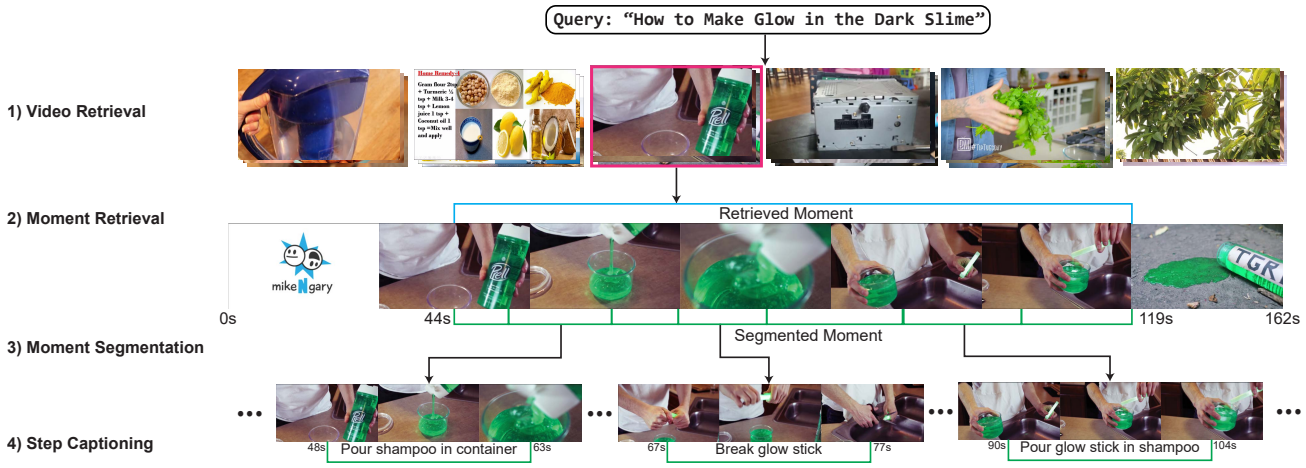

Figure 1. Overview of four hierarchical tasks of our HIREST dataset (Sec. 3). 1) Video retrieval: find a video that is most relevant to a given text query. 2) Moment retrieval: choose the relevant span of the video, by trimming the parts irrelevant to the text query. 3) Moment segmentation: break down the span into several steps and identify the start-end boundaries of each step. 4) Step captioning: generate step-by-step textual summaries of the moment.

erarchical video information retrieval.

Using the HIREST dataset, we benchmark four tasks: 1) video retrieval, 2) moment retrieval, 3) moment segmentation, and 4) step captioning (see Fig. 1 and Sec. 3.3). In the video retrieval task, models have to identify a video that is most relevant to a given text query. In the moment retrieval task, models have to select the relevant span of the video, by trimming the parts irrelevant to the text query (blue boundary in Fig. 1). In the moment segmentation task, models have to break down the relevant portion into several instructional steps and identify the start-end boundaries of each step (green boundaries in Fig. 1). Finally, in the step captioning task, models have to generate step captions (*e.g. 'spray the warm water on carpet'*) of the instructional steps. To provide good starting points to the community for our new task hierarchy, we show the performance of recent baseline models on HIREST. For baselines, we use strong models including CLIP [27], EVA-CLIP [8], Frozen-in-Time [2], BMT [13], and SwinBERT [20]. On all four tasks, we find that finetuning models on HIREST improve performance; however, there exists a large room to improve performance.

We summarize our contributions in this paper: 1) We present HIREST dataset and propose a new benchmark that covers hierarchy in information retrieval and visual/textual summarization from an instructional video corpus. 2) Unlike existing video datasets with step captions based on predefined task names or limited to a single topic, our HIREST provides diverse, high-quality step captions with timestamps written by human annotators. 3) We provide a joint baseline model that can perform moment retrieval, moment segmentation, and step captioning with a single architecture. 4) We provide comprehensive dataset analyses and show experiments with baseline models for each task, where there is a large room to improve model performance. We hope that HIREST can foster future work on end-to-end systems for holistic information retrieval and summarization on large video corpus. In addition, our manually annotated step captions can also be a good source for training and testing the step-by-step reasoning of large multimodal language models [40, 44].

2. Related Work

2.1. Text-based Information Retrieval from Video

With growing interest in building machine learning systems to search for useful information from large video corpora via text searches, several lines of work have been proposed. In text-to-video retrieval, a system finds the most relevant videos from a list of videos with a given text query [2, 17, 18, 41, 42]. In moment retrieval, a system finds the most relevant moments (usually a few seconds of frame spans), from a single video [10, 16, 17]. In query-focused video summarization, which is a text-conditional version of generic video summarization [9, 34], a system finds the most relevant frames from a video with text query [24, 33]. In video captioning, a system generates a short textual description of a given video [13, 20, 41, 42]. While all these tasks share common goals, information retrieval and summarization from a video corpus, previous works have focused on systems that are specialized in a single task. In this work, we introduce a holistic setup that combines video retrieval, moment retrieval, query-focused video summarization (called moment segmentation), and generating a step-wise textual summary of short clip (called step captioning), so that users can search for the most relevant video, the most

relevant moment inside the video, and get the stepwise text summarization of the moment.

2.2. Instructional Video Datasets

Recently, there have also been several efforts towards creating instructional video datasets [15, 23, 31, 36, 38, 45, 46]. While many of these datasets do a good job of providing high-quality instructional videos, they primarily only target a single domain [15, 31, 38, 45]. There have been recent strong efforts towards developing more diverse instructional datasets [23, 36, 46]. Datasets like HowTo100M [23] provide diverse instructional videos but lack specific step-by-step annotations. Some previous works such as [36, 46] provide step-level annotations for open domain videos, however, are restricted to a set of predefined steps that are reapplied across several videos. Our HIREST dataset provides step annotations on diverse instructional videos, where all step captions are manually written to answer the input text query by human annotators (see Table 1).

3. HIREST: Hierarchical Retrieval and Step-Captioning Dataset

We present HIREST, a video dataset consisting of 3.4K text-video pairs, 1.8K moments, and 8.6K step caption annotations. It covers the hierarchy of video/moment retrieval and stepwise captioning from a diverse instructional video corpus. Previous step annotations in video datasets used predefined task descriptions with small vocabulary [36, 46] or limited to a single domain (*e.g.* cooking [45]). In contrast, the step captions of HIREST are manually written by a human annotator and cover diverse domains with a large vocabulary (see Table 1). We describe the data collection process (Sec. 3.1), dataset analysis (Sec. 3.2), and four hierarchical tasks that stem from our dataset (Sec. 3.3).

3.1. Dataset Collection

In the following, we describe the two-stage data collection process. In the appendix, we provide screenshots of the data collection interface for each stage and worker qualification process.

Stage 1: Video and Moment Retrieval. We collect the pairs of text queries and relevant videos from the HowTo100M [23] dataset. Since videos were originally automatically collected from YouTube, we ensure that all videos are actually relevant to the query through human annotation. We employ crowdworkers from Amazon Mechanical Turk[2] and ask them to label whether or not the video correctly answers/solves the associated text query.

If the video is labeled as relevant to the text query, then we collect relevant 'moment' annotation from the video, by asking the crowdworkers to trim the video to the parts that are directly associated with the text (*i.e.* remove video parts unrelated to the text query, such as intro or other topics). We define a video as *clippable* to a moment, if the moment relevant to the query is less than 75% of the original video length. A system that can retrieve moments from videos would help people directly watch the video portion they are interested in and save time. For the retrieved moments, we collect more fine-grained annotations by dividing the moment into steps and captioning each step. We explain the moment annotation below.

Stage 2: Moment Segmentation and Step Captions. In this stage, we collect fine-grained, stepwise annotations of the retrieved moments. We ask crowdworkers to watch retrieved moments, divide them into several steps and mark the start timestamp of each step. Then, for each of the marked moment segments, they are asked to write a *step caption* that describes the specific step to complete (*e.g.* "add crayons to the candle", "melt it in bowl with hot water", "stir it well until dry"). Our text queries from HowTo100M [23] are instructional questions starting with "how to", and we want the step captions to serve as short textual summaries of moments/steps. we ask crowdworkers to start each caption with an action verb (*e.g.* "add", "apply") and limit the length of the captions to seven words.

3.2. Dataset Analysis

Task Category Distribution. Our videos and text queries are collected from the HowTo100M [23] dataset, and hence our category labels match theirs. As shown in Fig. 2, the most frequently occurring categories (for all text-video pairs and just videos with step captions) are "Hobbies and Crafts", "Food and Entertaining", and "Home and Garden". While these are the most common categories (similar to HowTo100M's most common categories), other categories still have a presence in our dataset.

Dataset Statistics. We collected a total of 3.4K text-video pairs, which are 287 seconds long on average, with a total duration of 270 hours. Out of 3.4K videos, 1.8K videos are *clippable* to a moment; *i.e.*, only a short clip (<75% of the original video) is relevant to the text query. The average moment length is 148 seconds, which is 55% of the original videos. Out of the 1.8K moments, we provide moment segmentation and step caption annotations for the randomly chosen 1.1K moments. The 1.1K moments are broken down to 7.6 steps on average, totaling 8.6K steps. Each step is annotated with a start-end timestamp and a step caption. The step captions are on average 4.42 words long and have 633 unique starting verbs with 3382 unique words. Fig. 4 shows the most frequent starting verbs and the most frequent words in the step captions (not counting the starting word and stop words). Fig. 3 shows the first three words of 50 random step caption samples (ignoring stop words). As

[2] https://www.mturk.com

Dataset	Domain	Step caption	# Videos / # Steps	# Steps per Moment	# Words per Caption	# Unique Captions	Avg. Duration (s) Video / Step
COIN [36]	Open	Predefined steps	11.8K / 46K	3.9	4.8	0.8K	142 / 14.9
CrossTask [46]	Open	Predefined steps	4.7K / 21K	7.4	2.4	0.1K	297 / 9.6
YouCook2 [45]	Cooking	Manually written	2K / 14K	7.7	8.8	13K	316 / 19.7
HIREST (Ours)	Open	Manually written	3.4K (1.1K w/ steps) / 8.6K	7.6	4.4	7.9K	263 / 18.9

Table 1. Comparison of HIREST and other video datasets with step annotations. While smaller in terms of the total number of videos than other datasets, HIREST covers various open-domain videos with many step annotations per video and high-quality step captions written by human annotators.

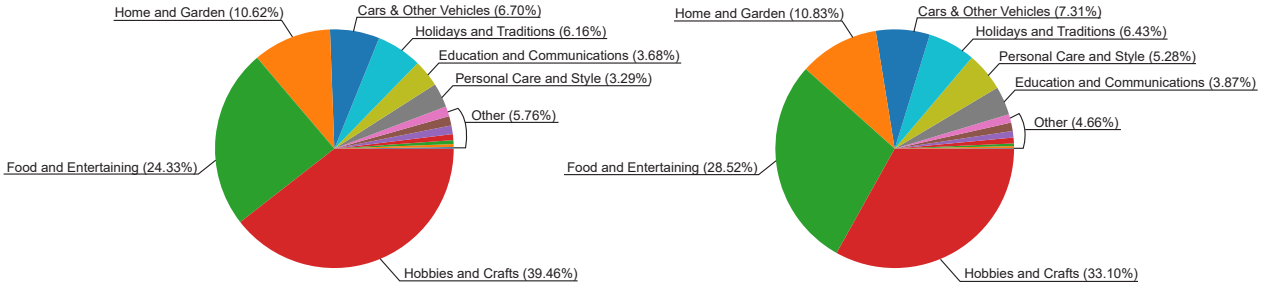

Figure 2. Task category distribution of HIREST text queries. There are a wide variety of categories for our videos. The most frequent categories are "Hobbies and Crafts", "Food and Entertaining", and "Home and Garden". The task categories are from HowTo100M [23].

shown in the visualizations, the manually written step captions of HIREST cover open domain instruction steps and have a diverse vocabulary.

Comparisons to Other Datasets with Step Captions. Table 1 compares our HIREST dataset to other video datasets with step annotations. HIREST covers various open-domain videos with many step annotations per video and high-quality step captions written by human annotations. While COIN [36] and CrossTask [46] also provide step-level annotations for open-domain videos, however, they are restricted to a set of predefined steps. In contrast, all the step captions of HIREST are manually written to answer the input text query.

Data Splits. Since there are cases where multiple videos are retrieved from the same query, we split our dataset into train/val/test splits by query instead of video. We split our queries into 546/292/546 (1507/477/1391 videos) for train/val/test splits, respectively.

3.3. Hierarchical Tasks Enabled by HIREST

In the following, we introduce four tasks connected in a hierarchy based on our HIREST dataset. See Fig. 1 for an overview and visual examples of the tasks.

Video Retrieval. This task gives models an instructional text query (*e.g.* "How to make a memory jar"), and the models need to determine which videos are relevant and retrieve the top results. The models must retrieve videos among

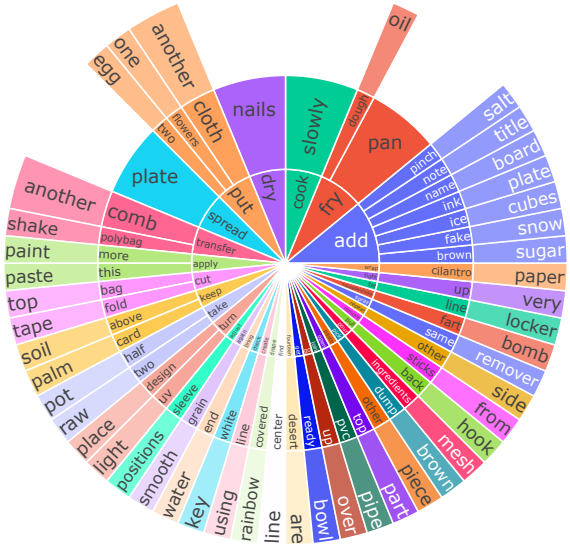

Figure 3. Distribution of HIREST step captions by their first three words for 50 random samples. Words are often related to actions or objects. We remove stop words (*e.g.* 'the', 'it', *etc.*).

4.2K test split videos (1.4K videos paired with text queries + 2.8K distractor videos from HowTo100M [23]). Distractor videos serve as negative examples (hence 'distractors'), similar to Revaud *et al.* [30]. We include these distractors to help increase the difficulty of our video retrieval task.

Moment Retrieval. In this task, the goal is to extract the

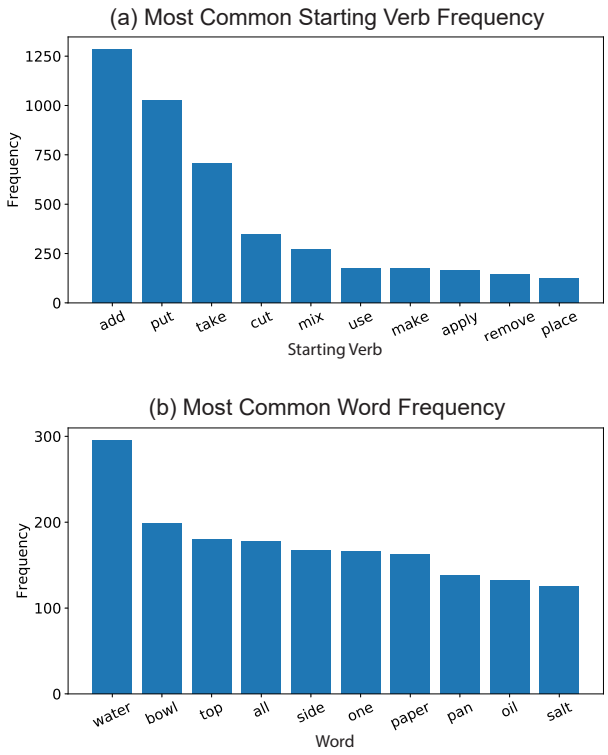

Figure 4. (a) Top 10 most common starting verbs in HIREST step captions. (b) Top 10 most common words in HIREST step captions (excluding the starting words and stop words). The top words typically refer to objects (*e.g.* water) or quantities (*e.g.* all).

portion of the video that is directly relevant to the given text query (i.e. to remove any unnecessary information from the start/end of the video).

Moment Segmentation. In this task, models should identify all relevant key 'steps' from the retrieved relevant moment of the video. Models should generate a list of start and end times for every key step in a given video.

Step Captioning. This task requires models to generate short textual step captions for each retrieved step in a video. Models are provided with the source video and start/end times of each step. They should then generate a short instructional step caption for every step.

4. Experiments

For all four HIREST tasks, we conduct experiments with task-specific baseline models (Sec. 4.1), a joint baseline model (Sec. 4.2), and evaluate them with different standard metrics (Sec. 4.3). We represent each video as 32 frames with uniform intervals, if not specified.

4.1. Task-specific Models

Video Retrieval. We experiment with CLIP (ViT-B/32) [27], EVA-CLIP (ViT-G/14) [8], Frozen-in-Time [2], and MIL-NCE (S3D) [22], which are pretrained text-to-image (CLIP/EVA-CLIP) and text-to-video (Frozen-in-Time/MIL-NCE) retrieval models, respectively. For CLIP and EVA-CLIP, we obtain a video embedding by averaging frame embeddings. We compute the matching score by taking the cosine similarity between video and text query embedding. Following the original setup, we use 4 frames for Frozen-in-Time and 32 frames for MIL-NCE.

Moment Retrieval. We experiment with two CLIP-based heuristics methods and the event proposal module of BMT [13], a dense video captioning model pretrained on ActivityNet Captions [14]. With CLIP, we compute the cosine similarity between all frames and the text query and find the frame with the highest score. Then we determine the start/end boundary of a moment with two different heuristics: 1) picking the frames where the similarity score drops from the highest scoring frame by a certain threshold (*e.g.*, 0.10); 2) picking the 8 frames to the left and right, totaling up to 17 (= 8+1+8) frames (see appendix for details). Furthermore, we experiment with the BMT [13] event proposal module, which predicts video event proposals with center/length/confidence values. We allow BMT to generate various events and then take the minimum start time and maximum end time across the events as the retrieved moment. For BMT, we give the model the I3D [5] RGB+Flow features and VGGish [11] audio features of the entire video, extracted at 1fps.

Moment Segmentation. We experiment with 1) frame-wise difference with the Structural Similarity Index Measure (SSIM) [39], and 2) the event proposal module of BMT [13]. For SSIM, if two adjacent frames have an SSIM below a certain threshold (*e.g.*, 0.85), we mark that as a step boundary. For BMT, we feed the model I3D and VGGish features (extracted at 1fps) of the entire video and directly use the video event proposal prediction.

Step Captioning.

We experiment with BMT and SwinBERT [20], a pretrained video captioning model. For BMT, we use I3D and VGGish features of each step, extracted at 1fps. We do not use its event proposal module for this task, as we give the features within the ground-truth step boundaries. For SwinBERT, we use YouCook2 [45] checkpoint and 32 video frames from each step as input to the model.

4.2. Joint Model

We also experiment with an end-to-end joint baseline model that handles moment retrieval, moment segmentation, and step captioning tasks with a single architecture. As shown in Fig. 5, our model is built on four ex-

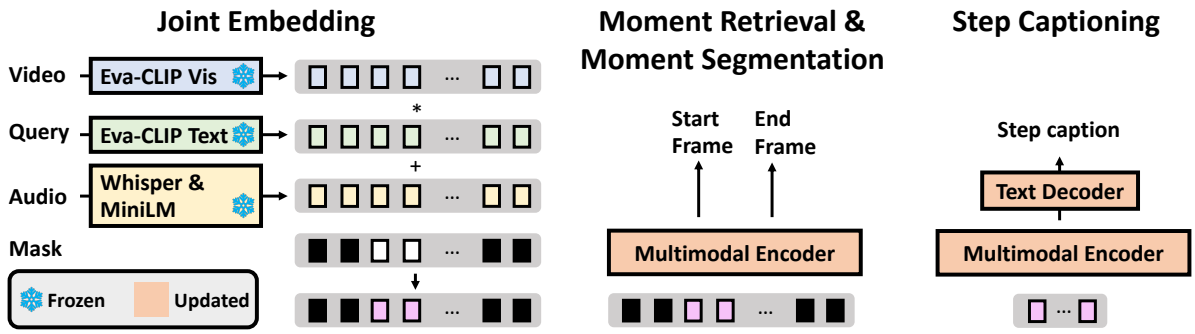

Figure 5. Illustration of our joint model that handles moment retrieval, moment segmentation, and step captioning tasks (Sec. 4.2). We learn a shallow multimodal transformer encoder layer that adapts the four pretrained models: EVA-CLIP (frozen), Whisper (frozen), MiniLM (frozen), and CLIP4Caption (finetuned).

isting pretrained models: EVA-CLIP [8], Whisper [28], MiniLM [29], and CLIP4Caption [35]. EVA-CLIP visual encoder maps a video frame into a visual embedding, EVA-CLIP text encoder maps a text query into a text embedding, Whisper extracts speech transcription from audio, MiniLM text encoder maps the speech transcription into a text embedding. To adapt the video, text, and audio embeddings, we finetune a two-layer multimodal encoder and a two-layer text decoder, which are initialized from CLIP4Caption (MSRVTT [41] checkpoint). We train the joint model in a multi-task setup in a round-robin fashion, by sampling a batch from one of the data loaders at each step [6].

Input Embedding. We construct the multimodal input embedding to the transformer by combining 1) EVA-CLIP video frame embedding, 2) EVA-CLIP text query embedding (tiled to the number of video frames), 3) and MiniLM speech transcription embedding (temporally warped into each frame), and 4) task-specific mask embeddings. For moment retrieval and moment segmentation tasks, we feed the same multimodal embeddings while masking out the frames that are outside of interest.

Moment Retrieval & Moment Segmentation. Following the span-based text question answering models [7, 32], we learn linear layers that predict the boundaries of moments and steps. Concretely, we use three linear layers predicting moment start, moment end, and step boundaries. For the moment retrieval, our joint start and end predictor predicts the moment boundary in parallel, and we do not mask out the video inputs. For the moment segmentation, our joint model autoregressively predicts each step's boundaries with masking; *i.e.*, we mask out 1) frames that are outside of the moment and 2) frames that are included in the previous steps. For both tasks, we feed the video in 1fps.

Step Captioning. Following CLIP4Caption [35], we sample 20 frames from each step. The autoregressive text decoder attends to the multimodal encoder output via cross-attention and generates each step caption independently.

Model	Frames	FT	R@1	R@5	R@10
CLIP-B/32	1		11.4	20.7	27.3
CLIP-B/32	4		12.5	28.8	37.4
CLIP-B/32	10		13.0	31.7	39.9
CLIP-B/32	20		13.0	33.3	41.2
CLIP-B/32	32		12.6	33.0	41.8
Frozen-in-Time	4		7.0	19.4	26.7
MIL-NCE (S3D)	32		13.9	31.1	41.4
CLIP-B/32	1	✓	11.5	22.7	27.1
CLIP-B/32	4	✓	13.9	29.5	39.4
CLIP-B/32	10	✓	11.4	31.3	41.4
CLIP-B/32	20	✓	12.3	31.7	41.6
CLIP-B/32	32	✓	13.0	32.1	41.9
EVA-CLIP-G/14	1		18.9	32.6	37.5
EVA-CLIP-G/14	4		20.7	43.6	53.7
EVA-CLIP-G/14	10		26.0	48.5	58.8
EVA-CLIP-G/14	20		**26.4**	**51.1**	**61.5**
EVA-CLIP-G/14	32		26.0	50.0	61.4

Table 2. Video retrieval results on HIREST test split. CLIP/EVA-CLIP results are based on temporal average pooling. *FT: finetuning on* HIREST*, R@k: Recall@k.* MIL-NCE was trained on the HowTo100M dataset, which is the video source of HIREST.

4.3. Metrics

Video Retrieval. Following previous work [2, 17–19, 42], We evaluate models on Recall@k metrics: R@1, R@5, and R@10.

Moment Retrieval. Following previous work [16, 17], we evaluate model outputs against the ground-truth (GT) moment spans with Recall@1 with Intersection over Union (IoU) thresholds (0.5 and 0.7).

Moment Segmentation. Following previous work [16, 17], we evaluate models on how similar the generated step spans are to the GT spans using IoU. We then compute the recall and precision with IoU thresholds (0.5 and 0.7).

Model	FT	R@0.5	R@0.7
CLIP-B/32 (threshold=0.05)		21.01	9.02
CLIP-B/32 (8 frames left/right)		34.02	15.72
EVA-CLIP-G/14 (threshold=0.10)		19.33	7.86
EVA-CLIP-G/14 (8 frames left/right)		38.27	19.33
BMT		43.56	10.57
BMT	✓	71.91	**39.18**
Joint (Ours)	✓	**73.32**	32.60

Table 3. Moment retrieval results on HIREST test split. CLIP (threshold): determines the start/end frames, by picking the frames where the similarity score drops from the highest scoring frame with a certain threshold (*e.g.*, 0.05). CLIP (8 frames left/right): determines the start/end frames by eight frames to the left and to the right of the highest scoring frame. *FT: Finetuning on* HIREST, *R@IoU: Recall@1 with a threshold of IoU*.

Model	FT	Recall@IoU		Precision@IoU	
		0.5	0.7	0.5	0.7
SSIM@0.75 (32 frames)		12.24	5.27	26.32	10.05
SSIM@0.85 (32 frames)		25.03	9.79	**37.38**	**13.80**
BMT (1fps)		8.24	3.71	20.95	7.96
BMT (1 fps)	✓	34.07	12.35	24.71	8.93
Joint (Ours) (1 fps)	✓	**37.50**	**14.76**	28.52	10.84

Table 4. Moment segmentation results on HIREST test split. We perform zeroshot evaluation with BMT, and then also provide results of using SSIM. SSIM is given 32 frames. *FT: Finetuning on* HIREST, *Recall/Precision@IoU: Recall@1/Precision with a threshold of IoU, SSIM@k: SSIM with a score threshold of k.*

Step Captioning. Following previous work [13, 19, 20, 41], we evaluate with the N-gram metrics: CIDEr [37], METEOR [3], and SPICE [1] with the language_evaluation package.[3] We also report two sentence-level embedding-based metrics BERTScore [43] and CLIPScore [12].

For BERTScore, we use the RoBERTa-Large [21]. For CLIPScore, we CLIP ViT-B/32 [27] and report the average of frame-caption cosine similarities using 4 frames uniformly sampled from each step. In addition, we compute the entailment of generated sentences to the GT sentences using the ELMo [26]-based Decomposable Attention model [25] pretrained on SNLI [4] with 3 labels: {`entailment`, `contradict`, `neutral`}.[4] We use the ratio of `entailment` prediction as the entailment score.

5. Results and Discussions

In the following, we present the experiment results on the four tasks and the visualization of the pipelined model predictions. Our baseline models show promising initial results, but there exists some gap between the current model performance and the upper bound accuracies, leaving large room for future improvements.

[3] https://github.com/bckim92/language-evaluation

[4] https://docs.allennlp.org/models/main/models/pair_classification/models/decomposable_attention/

Model	FT	METEOR	CIDEr	SPICE	Entail. (%)	BERT-S	CLIP-S
BMT		2.23	1.04	1.41	1.17	0.83	0.21
SwinBERT		5.12	13.31	4.65	5.86	0.85	**0.23**
BMT	✓	3.84	6.72	1.05	30.68	0.82	0.20
SwinBERT	✓	**5.94**	**24.66**	**6.67**	35.09	**0.86**	**0.23**
Joint (Ours)	✓	4.13	23.01	3.54	**43.88**	**0.86**	**0.23**

Table 5. Step captioning results on HIREST test split. We finetune each model on HIREST and evaluate them on our test split. *FT: Finetuning on* HIREST, *Entail: Entailment, BERT-S: BERTScore, CLIP-S: CLIPScore*.

Video Retrieval. Table 2 shows the video retrieval results. Increasing input frames increases the recall until 20 frames. Although CLIP was not trained on a video dataset, CLIP outperforms Frozen-in-Time (4 frames) shows comparable performance with MIL-NCE (32 frames). This is likely due to the fact that CLIP was trained on a much larger dataset than Frozen-in-Time. Finetuning CLIP on HIREST does not show a big difference. EVA-CLIP, a larger CLIP architecture with 1B parameters, outperforms all the other models with a big margin. Thus, we use EVA-CLIP as our video retrieval model and use its features for the three downstream tasks for our joint model.

Moment Retrieval. Table 3 shows the results for moment retrieval. Among the cosine similarity-based zero-shot methods, the 8-frame left/right method outperforms the similarity score drop difference method for both CLIP and EVA-CLIP. BMT achieves better R@0.5 than the zero-shot methods, and the finetuning improves both recall metrics. Our joint model outperforms finetuned BMT on the R@0.5, while finetuned BMT achieves a higher score on R@0.7.

Moment Segmentation. Table 4 shows the results for the moment segmentation task. In the zero-shot setting, BMT fails to adapt the span distribution of HIREST, and simple SSIM methods could outperform the BMT model on both recall and precision. But after finetuning, BMT shows significant improvement over its zero-shot version and SSIM methods on recall metrics. Our joint model achieves a better performance than BMT on both recall and precision.

Step Captioning. Table 5 shows the results of the step captioning task. For both BMT and SwinBERT, zero-shot inference did not result in a good result in N-gram (*e.g.*, CIDEr) and entailment metrics, indicating the domain gap between their pretraining datasets (ActivityNet caption and YouCook2) and HIREST is not negligible. For example, their captions are longer than step captions of HIREST. Finetuning brings a performance boost to BMT and SwinBERT in N-gram and entailment metrics but not in sentence-level embedding-based metrics (BERTScore and

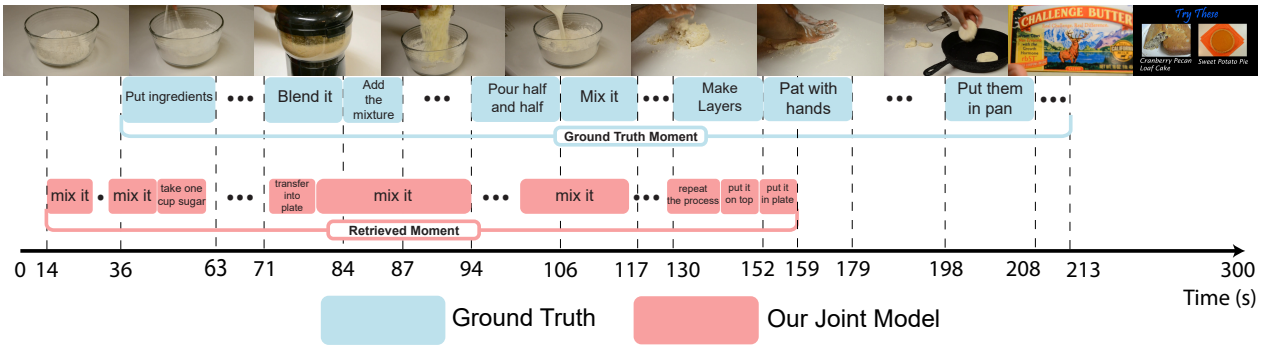

Figure 6. Comparison of our joint model prediction and ground truth annotation for moment retrieval, moment segmentation, and step captioning. The video is paired with a text query 'How to make butter biscuits'.

Model	FT	Moment Retrieval		Moment Segmentation		Step Captioning
		R@0.5	R@0.7	R@0.7	P@0.7	CIDEr
With Audio						
BMT	✓	71.9	39.2	12.4	8.9	6.7
Joint (Ours)	✓	73.3	32.6	14.8	10.8	23.0
Without Audio						
BMT	✓	62.6 (-9.3)	32.34 (-6.8)	10.4 (-2.0)	7.4 (-1.6)	6.1 (-0.6)
Joint (Ours)	✓	70.7 (-2.6)	20.6 (-12.0)	13.5 (-1.3)	10.0 (-0.8)	15.2 (--7.8)

Table 6. Ablation of using audio inputs. Removing audio input drops the performance of all three tasks for both models.

CLIPScore). Compared to SwinBERT, our joint model achieves similar CIDEr and sentence-level embedding-based metrics. Notably, our joint model outperforms SwinBERT significantly on the entailment metric. Future work on our dataset can also hopefully explore the complementary strengths of SwinBERT and our joint model.

Audio Ablation. Table 6 shows the ablation study about using (top rows) and not using (bottom rows) audio input with BMT and our joint model. Overall, both models show a performance drop without audio input. For moment retrieval and moment segmentation, removing audio input significantly drops the scores for both models, indicating that audio is very helpful for the tasks that require models to detect the boundaries of events. For the step captioning task, removing audio input significantly drops the score for our joint model, while BMT does not show a big difference.

Visualization of Hierarchical Model Pipelining. In Fig. 6, we visualize the model prediction results and ground-truth annotation for moment retrieval, moment segmentation, and step captioning tasks on a video associated with a query 'How to make butter biscuits'. The retrieved moment matches with the video moment about making the batter (36-159s) with the ground truth (GT) annotations. The predicted step boundaries and step captions also show semantic correspondence with GT annotations and the video. For example, the predicted caption 'mix it' matches the GT captions 'add the mixture' (84-87s) and 'mix it' (106-117s). The model also captions 'take one cup sugar' during that part where ingredients are added (47-55s). The model makes mistakes by missing the end of the dough cutting and the final cooking process (160-213s) during moment retrieval. In this period, we find that a human instructor stands up and describes the process, making the frames visually very different from the previous batter-making process.

6. Conclusion

In this work, we present the HIREST dataset and propose a new benchmark that covers hierarchy in information retrieval and summarization from an instructional video corpus. Our benchmark consists of four tasks: video retrieval, moment retrieval, and our new moment segmentation and step captioning tasks. Different from existing video datasets with step captions, our HIREST provides unique, diverse, high-quality instruction steps with timestamps written by human annotators. We provide comprehensive dataset analysis and present experiments with several task-specific and end-to-end joint baseline models for each task as starting points. We hope that HIREST can foster future work on multimodal systems for holistic video information retrieval, summarization, and step-by-step reasoning.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported by Meta AI, ARO Award W911NF2110220, DARPA KAIROS Grant FA8750-19-2-1004, and NSF-AI Engage Institute DRL-211263. The views, opinions, and/or findings contained in this article are those of the authors and not of the funding agency. For this work, the collection of data and the subsequent experiments were performed by the University of North Carolina at Chapel Hill, and not by Meta AI. As a result, both the data and code will be released by UNC Chapel Hill.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 7

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021. 1, 2, 5, 6

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, 2005. 7

[4] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015. 7

[5] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 5

[6] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying Vision-and-Language Tasks via Text Generation. In *ICML*, feb 2021. 6

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, oct 2019. 6

[8] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023. 2, 5, 6

[9] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014. 1, 2

[10] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. In *ICCV*, pages 5804–5813, 2017. 1, 2

[11] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. Cnn architectures for large-scale audio classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017. 5

[12] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In *EMNLP*, 2021. 7

[13] Vladimir Iashin and Esa Rahtu. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In *British Machine Vision Conference (BMVC)*, 2020. 1, 2, 5, 7

[14] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning Events in Videos. In *ICCV*, 2017. 5

[15] Hilde Kuehne, Ali Bilgin Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–787, 2014. 3

[16] Jie Lei, Tamara L. Berg, and Mohit Bansal. Qvhighlights: Detecting moments and highlights in videos via natural language queries. In *NeurIPS*, 2021. 1, 2, 6

[17] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 2020. 1, 2, 6

[18] Linjie Li, Yen-Chun Chen, Zhe Gan Yu Cheng, Licheng Yu, and Jingjing Liu. HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training. In *EMNLP*, 2020. 1, 2, 6

[19] Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, Tamara Lee Berg, Mohit Bansal, Jingjing Liu, Lijuan Wang, and Zicheng Liu. VALUE: A Multi-Task Benchmark for Video-and-Language Understanding Evaluation. In *NeurIPS*, pages 1–21, 2021. 6, 7

[20] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *CVPR*, 2022. 1, 2, 5, 7

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. 7

[22] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 5

[23] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019. 1, 3, 4

[24] Medhini Narasimhan, Arsha Nagrani, Chen Sun, Michael Rubinstein, Trevor Darrell, Anna Rohrbach, and Cordelia Schmid. Tl;dw? summarizing instructional videos with task relevance & cross-modal saliency. In *ECCV*, volume abs/2208.06773, 2022. 1, 2

[25] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*, 2016. 7

[26] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018. 7

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *ArXiv*, abs/2103.00020, 2021. 2, 5, 7

[28] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. 6

[29] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Lan-*

guage Processing. Association for Computational Linguistics, 11 2019. 6

[30] Jérôme Revaud, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, 2013. 4

[31] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1194–1201, 2012. 3

[32] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. Bi-Directional Attention Flow for Machine Comprehension. In *ICLR*, pages 1–12, 2017. 6

[33] Aidean Sharghi, Jacob S. Laurel, and Boqing Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *CVPR*, pages 2127–2136, 2017. 1, 2

[34] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, pages 5179–5187, 2015. 1, 2

[35] Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, DIan Li, and Xiu Li. CLIP4Caption: CLIP for Video Caption. In *ACM MM*, 2021. 6

[36] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 4

[37] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015. 7

[38] Weiying Wang, Yongcheng Wang, Shizhe Chen, and Qin Jin. YouMakeup: A large-scale domain-specific multimodal dataset for fine-grained semantic comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5133–5143, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 3

[39] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. 2

[41] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*, 2016. 1, 2, 6, 7

[42] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, volume 2017-Janua, pages 3261–3269, 2017. 1, 2, 6

[43] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In *ICLR*, 2019. 7

[44] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models, 2023. 2

[45] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 1, 3, 4, 5

[46] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3532–3540, 2019. 1, 3, 4