TL;DW? Summarizing Instructional Videos with Task Relevance & Cross-Modal Saliency

Medhini Narasimhan[1,2*], Arsha Nagrani[2], Chen Sun[2,3], Michael Rubinstein[2], Trevor Darrell[1†], Anna Rohrbach[1†], and Cordelia Schmid[2†]

[1]UC Berkeley [2]Google Research [2]Brown University

https://medhini.github.io/ivsum

Fig. 1: **Summarizing Instructional Videos** We introduce an approach for creating short visual summaries comprising steps that are most relevant to the task, as well as salient in the video, i.e. referenced in the speech. For example, given a long video on *"How to make a veggie burger"* shown above, the summary comprises key steps such as *fry ingredients, blend beans*, and *fry patty*.

Abstract. YouTube users looking for instructions for a specific task may spend a long time browsing content trying to find the right video that matches their needs. Creating a visual summary (abridged version of a video) provides viewers with a quick overview and massively reduces search time. In this work, we focus on summarizing *instructional* videos, an under-explored area of video summarization. In comparison to generic videos, instructional videos can be parsed into semantically meaningful segments that correspond to important steps of the demonstrated task. Existing video summarization datasets rely on manual frame-level annotations, making them subjective and limited in size. To overcome this, we first automatically generate *pseudo summaries* for a corpus of instructional videos by exploiting two key assumptions: (i) relevant steps are likely to appear in multiple videos of the same task (*Task Relevance*), and (ii) they are more likely to be described by the demonstrator verbally (*Cross-Modal Saliency*). We propose an instructional video summarization network that combines a context-aware temporal video encoder and a segment scoring

TL;DW? - Too Long; Didn't Watch?

*Work done while an intern at Google Research. Correspondence to medhini@berkeley.edu

†Equal contribution.

transformer. Using pseudo summaries as weak supervision, our network constructs a visual summary for an instructional video given only video and transcribed speech. To evaluate our model, we collect a high-quality test set, *WikiHow Summaries*, by scraping WikiHow articles that contain video demonstrations and visual depictions of steps allowing us to obtain the ground-truth summaries. We outperform several baselines and a state-of-the-art video summarization model on this new benchmark.

1 Introduction

The search query *"How to make a veggie burger?"* on YouTube yields thousands of videos, each showing a slightly different technique for the same task. It is often time-consuming for a first-time burger maker to sift through this plethora of video content. Imagine instead, if they could watch a compact visual summary of each video which encapsulates all semantically meaningful steps relevant to the task. Such a summary could provide a quick overview of what the longer video has to offer, and may even answer some questions about the task without the viewer having to watch the whole video. In this work, we propose a method to create such succinct visual summaries from long instructional videos.

Since our goal is to summarize videos, we consider prior work on generic [9,35] and query-focused [34] video summarization. Generic video summarization datasets [9,35] tend to contain videos from *unrestricted domains* such as sports, news and day-to-day events. Given that annotations are obtained manually, the notion of what constitutes a good summary is subjective, and might differ from one annotator to the next. Query-focused video summarization partially overcomes this subjectivity by allowing users to customize a summary by specifying a natural language query [34,22]. However, both generic and query-focused approaches require datasets to be annotated manually at a per-frame level. This is very expensive, resulting in very small-scale datasets (25-50 videos) with limited utility and generalization.

Here, we focus on a specific domain – that of instructional videos [36,45,21]. We argue that a unique characteristic of these videos is that a summary can be clearly defined as a minimally sufficient *procedural* one, i.e., it must include the steps necessary to complete the task (see Fig. 1). To circumvent having to manually annotate our training data, we use an unsupervised algorithm to obtain weak supervision in the form of pseudo ground-truth summaries for a large corpus of instructional videos. We design our unsupervised objectives based on two hypotheses: (i) steps that are relevant to the task will appear across multiple videos of the same task, and (ii) salient steps are more likely to be described by the demonstrator verbally. In practice, we segment the video and group individual segments into steps based on their visual similarity. Then we compare the steps across videos of the same task to obtain *task relevance scores*. We also transcribe the videos using Automatic Speech Recognition (ASR) and compare the video segments to the transcript. We aggregate these *task relevance* and *cross-modal scores* to obtain the *importance scores* for all segments, i.e., our pseudo ground-truth summary.

Next, given an input video and transcribed speech, we train an instructional video summarization network (*IV-Sum*). IV-Sum learns to assign scores to short *video segments* using 3D video features which capture temporal context. Our network consists of a video encoder that learns context-aware temporal representations for each segment and a segment scoring transformer (SST) that then assigns importance scores to each segment. Our model is trained end-to-end using the importance scores from the pseudo summaries. Finally, we concatenate the highest scoring segments to form the final video summary.

While we can rely on pseudo ground-truth for training, we collect a clean, manually verified test set to evaluate our method. Since manually creating a labeled test set from scratch would be extremely expensive, we find a solution in the form of the WikiHow resource[1]. WikiHow articles often contain a link to an instructional video and a set of human-annotated steps present in the task along with corresponding images or short clips. To construct our test set (referred to as *WikiHow Summaries*), we automatically localize these images/clips in the video. We obtain localized segments for the images (using a window around the localized frame) and clips, and stitch the segments together to create a summary. This provides us with binary labels for each frame which serve as ground-truth annotations. We evaluate our model on *WikiHow Summaries* and compare it to several baselines and the state-of-the-art video summarization model CLIP-It [22]. Our model surpasses prior work and several baselines on three standard metrics (F-Score, Kendall [15], and Spearman [46] coefficients).

To summarize (pun intended), we introduce an approach for summarizing instructional videos that involves training our *IV-Sum* model on pseudo summaries created from a large corpus of instructional videos. *IV-Sum* learns to rank different segments in the video by learning context-aware temporal representations for each segment and a segment scoring transformer that assigns scores to segments based on their task relevance and cross-modal saliency. Our method is weakly-supervised (it only requires the task labels for videos), multimodal – uses both video and speech transcripts, and is scalable to large online corpora of instructional videos. We collect a high-quality test set, *WikiHow Summaries* for benchmarking instructional video summarization, which will be publicly released. Our model outperforms state-of-the-art video summarization methods on all metrics. Compared to the baselines, our method is especially good at capturing task relevant steps and assigning higher scores to salient frames, as seen through qualitative analysis.

2 Related Work

We review several lines of work related to summarization of instructional videos.
Generic Video Summarization. This task involves creating abridged versions of generic videos by stitching together short important clips from the original video [10,19,22,25,29,40,42,43,44]. Some of the more recent methods attempt to learn contextual representations to perform video summarization, via attention

[1] https://www.wikihow.com/

mechanism [7], graph based [25] or transformer-based [22] methods. Representative datasets include SumMe [9] and TVSum [35], where the ground-truth summaries were created by annotators assigning scores to each frame in the video, which is highly time consuming and expensive. As a consequence, the generic video summarization datasets are small and the quality of the summaries is often very subjective. Here, we focus on instructional videos which contain structure in the form of task steps, thus we have a clear definition of what a good summary should contain - a set of necessary steps for performing that specific task.

Query Focused Video Summarization. To address the subjectivity issues with Generic Summarization, Query Focused Video Summarization allowed for having user defined natural language queries to customize the summaries [14,34,38]. A representative dataset is Query Focused Video Summarization [33]; it is very small and the queries correspond to a very narrow set of objects. In contrast, our task is large and we do not rely on any additional user input.

Step Localization. Step localization (also known as temporal action segmentation) is a related albeit distinct task. It typically implies predicting temporal boundaries of steps when the step labels [28,36,45] and even their ordering [2,4,6,12,17,27] are given. Representative datasets, COIN [36] and CrossTask [45] consist of instructional videos and a fixed set of steps for each task (from the WikiHow resource), and the task is to localize these steps in the video. Our task is different in that we are only given a video without corresponding input steps. Our model learns to pick out segments that correspond to relevant and salient steps in order to construct a video summary. We discuss and illustrate the shortcomings of the step localization annotations in Sec. 5 and Fig. 6.

Unsupervised Parsing of Instructional Videos. Closest to ours is the line of work on unsupervised video parsing and segmentation that discovers steps in instructional videos in an unsupervised manner [1,8,18,32,31]. However, these works - (1) do not focus on video summarization, thus they might miss some salient steps in video, (2) often use very small datasets for training and evaluation that do not capture the broad range of instructional videos found in, e.g., COIN [36] and CrossTask [45].

3 Summarizing Instructional Videos

Overview. We propose a novel approach for constructing visual summaries of instructional videos. An instructional video typically consists of a visual demonstration of a specific task, e.g. *"How to make a pancake?"*. Our goal is to construct a visual summary of the input video containing only the steps that are crucial to the task and salient in the video, i.e. referenced in the speech. Fig. 2 illustrates an outline of our approach. Our instructional video summarization pipeline consists of two stages - (i) first, we use a weakly supervised algorithm to generate pseudo summaries and frame-wise importance scores for a large corpus of instructional videos, relying only on the task label for each video (ii) next, using the pseudo summaries as supervision, we train an instructional video summarization network which takes as input the video and the corresponding

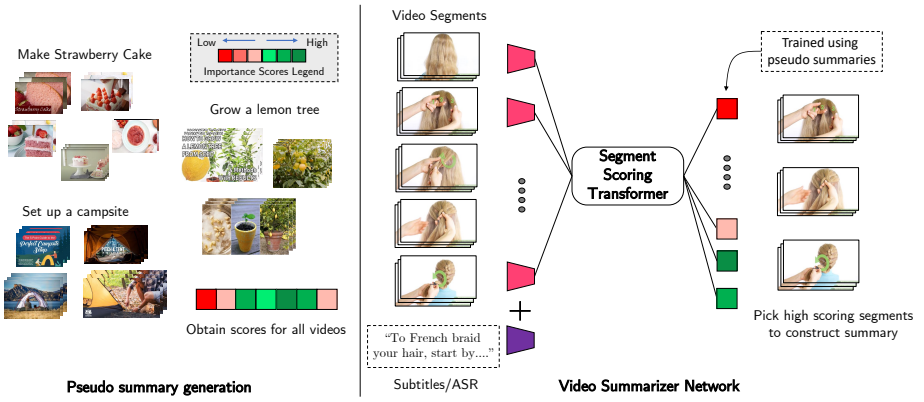

Fig. 2: **Summarizing Instructional Videos.** We first obtain pseudo summaries for a large collection of videos using our weakly supervised algorithm (more details in Fig. 3). Next, using the pseudo summaries as weak-supervision, we train our Instructional Video Summarizer (*IV-Sum*). It takes an input video along with the corresponding ASR transcript and learns to assign importance scores to each segment in the video. The final summary is a compilation of the high scoring video segments.

transcribed speech and learns to assign scores to different segments in the input video. The network consists of a video encoder and a segment scoring transformer (SST) and is trained using the importance scores of the pseudo summaries. The final summary is constructed by selecting and concatenating the segments with high importance scores. We first describe our pseudo summary generation algorithm, followed by details on our instructional video summarizer (*IV-Sum*), and the inference procedure.

3.1 Generating Pseudo Summaries

Since manually collecting annotations for summarization is expensive and time consuming, we propose an automatic weakly supervised approach for generating summaries that may contain noise but have enough valuable signal for training a summarization network. The main intuition behind our pseudo summary generation pipeline is that given many videos of a task, steps that are crucial to the task are likely to appear across multiple videos (task relevance). Additionally, if a step is important, it is typical for the demonstrator to speak about this step either before, during, or after performing it. Therefore, the subtitles for the video obtained using Automatic Speech Recognition (ASR) will likely reference these key steps (cross-modal saliency). These two hypotheses shape our objectives for generating pseudo summaries.

Task Relevance. We first group videos based on the task. Say videos $V_i, i \in [1, \dots \mathcal{K}]$ are $\mathcal{K}$ videos from the same task, as shown in Fig. 3. For a given video, we divide it into $\mathcal{N}$ equally sized non-overlapping segments $s_i, i \in [1, \dots \mathcal{N}]$ and embed each segment using a pre-trained 3D CNN video encoder g_{vid} [20]. We

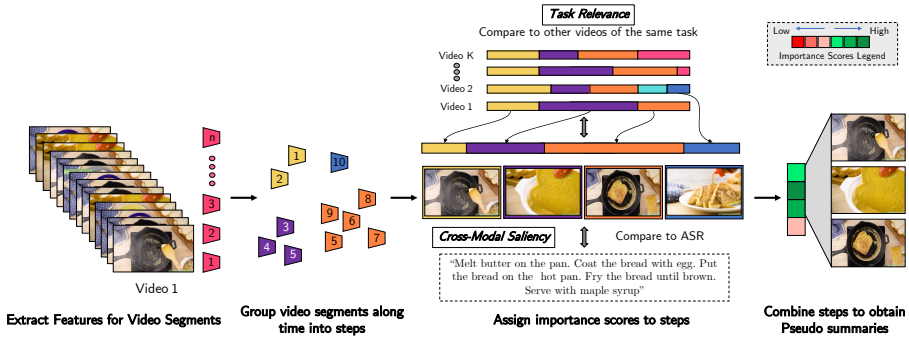

Fig. 3: **Pseudo Summary Generation.** To generate the pseudo summary, we first uniformly partition the video into segments, then group the segments based on visual similarity into steps (shown in different colors), assign *importance scores* to steps based on *Task Relevance* and *Cross-Modal Saliency*, and then pick high scoring steps to obtain pseudo summaries.

merge segments along the time axis based on their dot-product similarity, i.e. if similarity of a segment to the one prior to it is greater than a threshold, the two are grouped together and the joint feature representation is an average of the feature representation of the two segments. The threshold for similarity is heuristically set to be 90% of the maximum similarity between any two segments in the video. We call these merged segments *steps*, as they typically correspond to semantic steps as we show through qualitative results in supplemental. We do this for all $\mathcal{K}$ videos in the task, and then compare each step to all the $\mathcal{S}$ steps across all $\mathcal{K}$ videos of the task. We assign *task relevance scores* trs_{S_i}, to each step $S_i, i \in \mathcal{S}$ based on its visual similarity to all the $\mathcal{S}$ steps from all K videos of this task, as shown below:

$$\text{trs}_{S_i} = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} g_{vid}(S_i) \cdot g_{vid}(S_j)$$

Cross-Modal Saliency. We also compare each video step to each sentence in the transcript of the same video. This enforces our idea that if a step is important, it will likely be referenced in the speech. To do this, we encode both, the input segments and the transcript sentences, using a pre-trained video-text model where the video and text streams are trained jointly using MIL-NCE loss [20]. Each visual step is assigned a *cross-modal score* by averaging its similarity over all the sentences.

Each step (and all the segments in it) is then assigned an importance score that is an average of the *task relevance* and the *cross-modal scores*. This constitutes our pseudo summary scores. For any given video, the top t% highest scoring steps are retained to be a part of the summary.

3.2 Instructional Video Summarizer (*IV-Sum*)

Recall that our goal is to construct a visual summary of any instructional video by picking out the important steps in it, without having to rely on other videos of the same task or the task label. To do this, we use the pseudo summaries generated above as weak supervision to train *IV-Sum*, which learns to assign importance scores to individual segments in the video using only the information in the video and the corresponding transcripts as seen in Fig. 2. While some prior summarization methods operate on independent frames [22,25], *IV-Sum* operates on non-overlapping segments $s_i, i \in [1, \ldots \mathcal{N}]$, and learns *context-aware temporal representations* using a 3D CNN video encoder f_{vid}. The transcript is projected onto the same embedding space using a text encoder f_{text}, and the text representations are concatenated individually to each of the segments. To contextualize information across several segments, we use a segment scoring encoder-only transformer [37] f_{trans} with positional embeddings, that assigns importance scores Y'_{s_i} to each segment as shown in Eq. 1. The network is trained using supervision from the importance scores of the pseudo summaries Y_{s_i}, using Mean-Squared Error Loss as shown in Eq. 2.

$$Y'_{s_i} = f_{\text{trans}}(\text{concat } (f_{\text{text}}(\text{transcript}), f_{\text{vid}}(s_i))) \ \forall \ i \in \mathcal{N} \tag{1}$$

$$\mathcal{L}_{\text{IV-Sum}} = \sum_{i \in \mathcal{N}} \text{MSE } (Y'_{s_i}, Y_{s_i}) \tag{2}$$

During inference, we sort the segments based on the predicted scores and assign the label 1 to the top t% of the segments, and the label 0 to the remaining ones. When a segment is assigned a label, all the frames in the segment also get assigned the same label. The summary is constructed by stitching together all the frames with label 1.

4 Instructional Video Summarization Datasets

We describe the details of the data collection process for the annotations used in our work — *Pseudo Summaries* annotations for training and the *WikiHow Summaries* annotations for evaluation.

Pseudo Summaries Training Dataset. As described in Sec. 3.1, we use the pseudo summary generation process for creating our training set. We use the videos and task annotations from COIN [36] and CrossTask [45] datasets for creating our training datasets.

COIN: COIN consists of 11K videos related to 180 tasks. As this is a dynamic YouTube dataset, we were able to obtain 8,521 videos at the time of this work.

Cross-Task: CrossTask consists of 4,700 instructional videos (of which we were able to access 3,675 videos) across 83 different tasks.

Pseudo Summaries: We combined the two datasets to create pseudo summaries comprising of 12,160 videos, whilst using the videos that were common to both datasets only once. They span 263 different tasks, have an average length of

Table 1: **Instructional Video Summarization Datasets Statistics.** † Our *WikiHow Summaries* dataset was created automatically using a scalable pipeline, but manually verified for correctness.

	TVSum	SumMe	Pseudo Summaries	WikiHow Summaries
Number of videos	50	25	12160	2106
Annotation	Manual	Manual	Automatic	Manually verified†
Number of Tasks/Categories	10	25	185	20
Total Input Duration (Hours)	3.5	1.0	628.53	42.94

3.09 minutes, and in total comprise of 628.53 hours of content. The summary videos that were constructed using our pseudo ground-truth generation pipeline are 1.71 minutes long on an average, with each summary being 60% of the original video. While it is possible to construct pseudo summaries using the step-localization annotations, we show in Sec. 5 that such summaries may miss important steps or do not pick up on steps that are salient in the video. Moreover, our pseudo summary generation mechanism is weakly-supervised, requiring only task annotations and no step-localization annotations.

WikiHow Summaries Dataset. To provide a test bed for instructional video summarization, we automatically create and manually verify *WikiHow Summaries*, a video summarization dataset consisting of 2,106 input videos and summaries, where each video describes a unique task. Each article on the WikiHow Videos website consists of a main instructional video demonstrating a task that often includes promotional content, clips of the instructor speaking to the camera with no visual information of the task, and steps that are not crucial for performing the task. Viewers who want an overview of the task would prefer a shorter video without all of the aforementioned irrelevant information. The WikiHow articles (e.g., see How to Make Sushi Rice) contain exactly this: corresponding text that contains all the important steps in the video listed with accompanying images/clips illustrating the various steps in the task. These manually annotated articles are a good source for automatically creating ground-truth summaries for the main videos. We obtain the summaries and the corresponding labels and importance scores using the following process (see supp. for an overview figure):

1. Scraping WikiHow videos. We scrape the WikiHow Videos website for all the long instructional videos along with each step and the images/video clips (GIFs) associated with the step.

2. Localizing images/clips. We automatically localize these images/clips in the main video by finding the closest match in the video. To localize an image, we compare ResNet50 [11] features of the image and to that of all the frames in the video. The most similar frame is selected and this step is localized in the input video to a 5 second window centered around the frame. If the step contains a video clip/GIF, we localize the first frame of the video clip/GIF in the input video by similarly comparing ResNet features, as above, and the localization is set to be the length of the step video clip.

3. Ground-truth summary from localized clips. We stitch the shorter localized clips together to create the ground truth summary video. Consequently, we assign labels to each frame in the input video, depending on whether it belongs to the input summary (label 1) or not (label 0). To obtain importance scores, we partition each input video into equally sized segments (same as in Sec. 3.2) and compute the importance score for each segment to be the average of the labels assigned to the individual frames in the segment.
4. Manual verification. We verified that the summaries are at least 30% of the original video and manually fixed summaries that were extremely short/long.
Online Longevity and Scalability. We note that a common problem plaguing YouTube datasets today is shrinkage of datasets as user uploaded videos are taken down by users (eg. Kinetics [3]). WikiHow articles are less likely to be taken down, and this is an actively growing resource as new How-To videos are released and added (25% growth since we collected the data). Hence there is a potential to continually increase the size of the dataset.

For each video, we provide the following: (i) frame-level binary labels (ii) the summary formed by combining the frames with label 1 (iii) segment-level importance scores between 0 and 1, which are computed as an average of the importance scores for all the frames in the segment (iv) the localization of the visual steps in the video (i.e. the frames associated with each step). We also scrape natural language descriptions of each step as a bonus that could be useful for future work. We divide our WikiHow dataset into 768 validation and 1,339 test videos. Tab. 1 shows the statistics of both our datasets. Both datasets are much larger in size compared to existing generic video summarization datasets, contain a broader range of tasks, and are scalable.

5 Experiments

Next, we describe the experimental setup and evaluation for instructional video summarization. We compare our method to several baselines, including CLIP-It [22], the state-of-the-art on generic and query-focused video summarization.
Implementation Details. For the video and text encoders, we use an S3D [39] network, initialized with weights from pre-training on HowTo100M [21] using the MIL-NCE loss [20]. We fine-tune the *mixed_5** layers and freeze the rest. The segment scoring transformer is an encoder consisting of 24 layers and 8 heads and is initialized randomly. The network is trained using the Adam optimizer [16], with learning rate of 0.01, and a batch size of 24. We use Distributed Data Parallel to train for 300 epochs across 8 NVIDIA RTX 2080 GPUs. Additional implementation details are mentioned in supplemental.
Metrics. To evaluate instructional video summaries, we follow the evaluation protocol used in past video summarization works [41,25,22] and report Precision, Recall and F-Score values. As described in Sec. 4, each video in the *WikiHow Summaries* dataset contains the ground-truth labels Y_l (binary labels for each frame in the video) and the ground-truth scores Y_s (importance scores in the range [0-1] for each segment in the video). We compare the binary labels predicted

Table 2: **Instructional Video Summarization results on *WikiHow Summaries*.** We compare F-Score, Kendall and Spearman correlation metrics of our method IV-Sum, to all the baselines. Our method achieves state-of-the-art on all three metrics.

Method				F-Score		τ [15]	ρ [46]
				Val	Test	Test	Test
	ASR	RGB	Pseudo				
Frame Cross-Modal Similarity	✓	✓	-	52.8	53.1	0.022	0.051
Segment Cross-Modal Similarity	✓	✓	-	55.1	55.5	0.034	0.060
Step Cross-Modal Similarity	✓	✓	-	57.9	58.3	0.037	0.061
CLIP-It with captions [22]	-	✓	-	22.5	22.1	0.036	0.064
CLIP-It with ASR [22]	✓	✓	-	27.9	27.2	0.055	0.088
CLIP-It with ASR	✓	✓	✓	62.5	61.8	0.093	0.191
IV-Sum without ASR	-	✓	✓	65.8	65.2	0.095	0.202
IV-Sum	✓	✓	✓	**67.9**	**67.3**	**0.101**	**0.212**

for the frames in the video Y_l', to the ground truth labels Y_l, and measure F-Score, Precision and Recall, as defined in prior summarization works [30,29].

While these scores assess the quality of the predicted frame-wise binary labels, to assess the quality of the predicted segment-wise importance scores Y_s', we follow Otani *et al.* [24], and report results on the rank-based metrics Kendall's τ [15] and Spearman's ρ [46] correlation coefficients. We first rank the video frames according to the generated importance scores Y_s' and the ground-truth importance scores Y_s. We then compare the generated ranking to each ground-truth ranking of video segments for each video obtained from the frame-wise binary labels as described in Sec. 4. The final correlation score is computed by averaging over the individual scores for each video.

Baselines. We compare our method to the state-of-the-art video summarization model CLIP-It [22]. To validate the need for pseudo summaries, we construct three unsupervised baselines as alternatives to our pseudo summary generation algorithm. We first describe the three unsupervised baselines.

Frame Cross-Modal Similarity. We sample frames (at the same FPS used by our method) from an input video and compute the similarity between CLIP (ViT-B/32) [26] frame embeddings and CLIP text embeddings of each sentence in the transcript. The embeddings do not encode temporal information but leverage the priors learned by the CLIP model. Based on the scores assigned to each frame, we threshold t% of the higher scoring frames to be part of the summary. Frame scores are propagated to the segments they belong to, and the summary is a compilation of the chosen segments.

Segment Cross-Modal Similarity. We uniformly divide the video into segments and compute MIL-NCE [20] video features for each segment. We embed each sentence in the transcript to the same feature space using the MIL-NCE text encoder. We compute the pairwise similarity between all video segments and the sentences, and average over sentences to obtain a score for each segment.

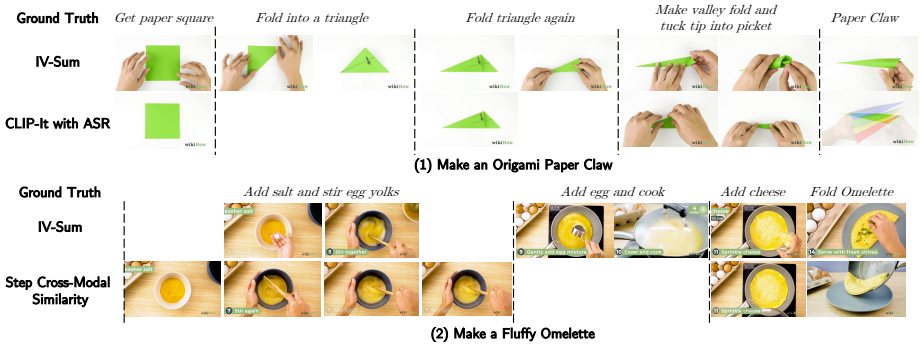

Fig. 4: **Qualitative comparisons to baselines.** We show the steps in the ground-truth as text (note we never train with step descriptions, these are shown here simply for illustrative purposes) and compare frames selected in summaries generated by our method IV-Sum, CLIP-It with ASR, and Step Cross-Modal Similarity. In (1), CLIP-It misses steps which are deemed important by our method (*"Fold into a triangle"*) and assigns higher scores to less salient frames for the step (*"Make valley fold and tuck tip into picket"*) where neither the valley fold nor the picket are clearly visible. In (2), Step Cross-Modal Similarity misses (*"Add egg and cook"*) and selects too many redundant frames for the step (*"Add salt and stir egg yolks"*).

Our intuition is that since demonstrators typically describe the important steps shortly before, after or while performing them, a high similarity between the visuals and transcripts would directly correlate with the significance of the step. We filter t% of the highest scoring segments, where t is determined heuristically using the *WikiHow Summaries* validation set and is consistent across all baselines and our model. The filtered segments are stitched together to form the summary.

Step Cross-Modal Similarity. We first group segments into steps and then compare them to the ASR transcripts. For this we employ the technique described in Sec. 3.1, i.e. we extract MIL-NCE features for the video segments and group them together based on their similarity to form steps.[2] The embedding for a step is set to be the average of all the segment embeddings in it. If a step is similar to the transcript, all the segments in that step are chosen to be part of the summary. This baseline is the closest to our pseudo summary generation algorithm.

Next, we describe the CLIP-It baseline and ablations, trained with supervision.

CLIP-It with captions. We evaluate CLIP-It [22] trained on TVSum [35], SumMe [9], OVP [23], and YouTube [5] against our *WikiHow Summaries*. We use the same protocol as in CLIP-It for evaluation and describe further details in supplemental. For language-conditioning, we follow CLIP-It and generate captions for the *WikiHow Summaries* dataset using BMT [13]; we feed these as input to the CLIP-It model.

[2] Since we process a single input video (not multiple videos per task), we can not use the Task Relevance component.

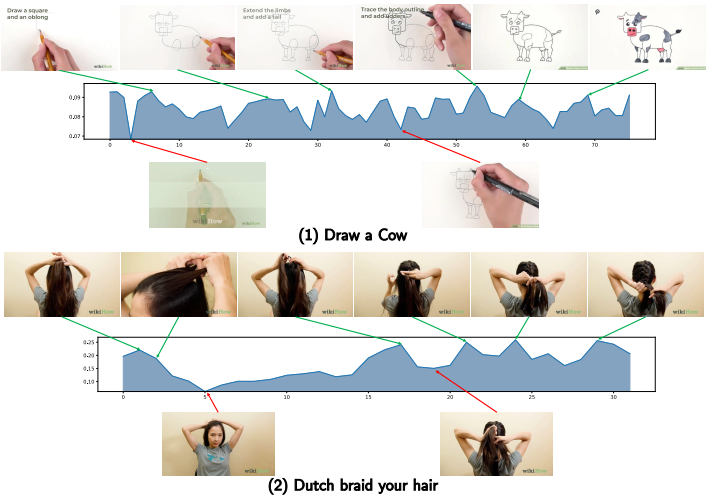

Fig. 5: **Qualitative results.** We show summaries from our method IV-Sum along with the predicted importance scores. The green and red arrows point to frames that were assigned a high and low scores, respectively. Our model correctly assigns higher scores to frames from all the steps that are relevant and lower scores to frames which aren't crucial to the task (as in (1)) and frames which don't belong to a step (as in (2)).

CLIP-It with ASR transcripts. We evaluate the same CLIP-It model above by replacing captions with ASR transcripts, so as to allow for a fair comparison with the baselines and our method, IV-Sum which use ASR transcripts.

CLIP-It with ASR transcripts trained on Pseudo Summaries. We train CLIP-It from scratch on our Pseudo-GT Summaries dataset using ASR transcripts from the videos in place of captions.

Quantitative Results. We compare the baselines to the two versions of IV-Sum, one with ASR transcripts and another without. To train IV-Sum without transcripts, we simply eliminate the text encoder (f_{text}) in Eq. 1 and pass only the visual embeddings of the individual segments to the transformer. We report F-Score, Kendall's τ and Spearman's ρ coefficients in Tab. 2. As seen, IV-Sum (both with and without ASR transcripts), outperforms all the baselines on all metrics. Particularly, we achieve notable improvements on the correlation metrics that compare the saliency scores, attesting to our model's capabilities to assign higher scores to segments that are more relevant. We also observe that CLIP-It trained using the pseudo summaries generated by our method has a strong boost in performance compared to CLIP-It trained on generic video summarization datasets, reinforcing the effectiveness of our pseudo summaries for training. The best method among the unsupervised ones is Step Cross-Modal Similarity, a "reduced" version of our pseudo summary generation method.

Qualitative Results. We present qualitative results in Fig. 4. We show frames in the summaries generated by our method IV-Sum, CLIP-It with ASR tran-

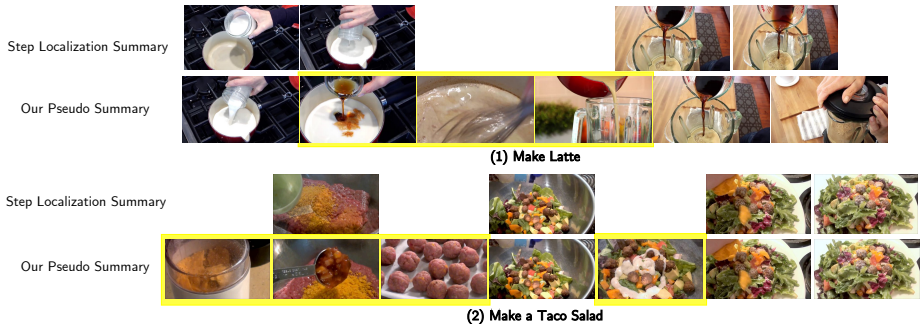

Fig. 6: **Pseudo summaries vs step-localization annotations.** We compare frames in our automated pseudo summary to the step localization manual annotations, aligned temporally. Frames corresponding to steps that are identified by our method but missed by step localization are highlighted in yellow.

scripts (trained on generic video summarization datasets), and Step Cross-Modal Similarity. We also list the steps in the ground-truth as text (for illustrative purposes). In Fig. 4 (1), CLIP-It misses the step *"Fold into a triangle"*, as it optimizes for diversity among the frames and was trained on a small dataset that does not generalize well to our domain. It also picks the less salient frames for the step *"Make valley fold and tuck tip into picket"*, whereas our model correctly identifies all the steps and assigns higher scores to the more salient frames. The summary from the Step Cross-Modal Similarity baseline, shown in Fig. 4 (2), assigns high scores to several redundant frames (*"Add salt and stir egg yolks"*), but misses "Add egg and cook".

Fig. 5 shows results from our method along with the predicted frame-wise importance scores. The green and red arrows point to frames that are assigned the highest and lowest scores by our method, respectively. As seen, our method assigns high scores to frames in task relevant and salient steps and low scores to frames which aren't crucial to the step, like in Fig. 5 (1), or do not belong to a step, like in Fig. 5 (2) where the person is talking to the camera.

Ablations. We compare different approaches to generate pseudo summaries for training our instructional video summarizer network – (i) First, we ablate the two objectives, Task Relevance and Cross-Modal Saliency, used to generate the pseudo summaries. (ii) Next, we replace the annotations from our pseudo summary generation pipeline with step localization annotations. We include model and loss ablations in the supplemental.

(i) Ablating Objectives. We ablate the two objectives, Task Relevance and Cross-Modal Saliency, used for generating pseudo summaries, in Tab. 3a. We train IV-Sum on different versions of pseudo summaries and report F-Scores on the *WikiHow Summaries* validation set. Combining both objectives is more effective than using each objective individually.

(ii) Using Step Localization Annotations. COIN and CrossTask datasets contain temporal localization annotations of a generic set of steps pertaining to the task in

Table 3: **Pseudo Summary Variations.** We report results on two variations of generating the pseudo summaries: (i) ablating the objectives (ii) using step localization annotations to generate pseudo summaries.

(a) **Ablating objectives.** We ablate the two objectives in our pseudo summary generation pipeline.

Method	**F-Score**
Task-Consistency only	64.1
Cross-Modal Similarity only	61.0
Both	67.9

(b) **Using Step-Localization Annotations.** We compare pseudo summaries from step-localization annotations with our approach.

Method	F-Score
IV-Sum (Step Localization)	57.6
IV-Sum (Ours)	66.8

the videos. We use these annotations to extract the visual segments corresponding to the steps and concatenate them to form a summary. We assign binary labels to each frame, depending on whether they belong in the summary or not. We then use these step-localization summaries as supervision to train our model, IV-Sum with a weighted-CE loss [22] as this works best for binary labels. In Tab. 3b, we compare this to IV-Sum trained on pseudo summaries generated using our pipeline and report F-Scores on our *WikiHow Summaries* validation set. As seen, IV-Sum trained on our generated summaries outperforms IV-Sum trained using step-localization summaries. We qualitatively compare our automatic pseudo summaries to the manually labeled step localization annotations in Fig. 6. Often the step annotations only cover a few steps and miss other crucial steps as shown in yellow in (1). In (2), we observe that our pseudo summary retrieves steps that are unique to the task which the step localization annotation doesn't include.

6 Conclusion

We introduce a novel approach for generating visual summaries of instructional videos — a practical task with broad applications. Specifically, we overcome the need to manually label data in two important ways. For training, we propose a weakly-supervised method to create pseudo summaries for a large number of instructional videos. For evaluation, we leverage WikiHow (its videos and step illustrations) to automatically build a *WikiHow Summaries* dataset. We manually verify that the obtained summaries are of high quality. We also propose an effective model to tackle instructional video summarization, IV-Sum, that uses temporal 3D CNN representations, unlike most prior work that relies on frame-level representations. We demonstrate that all components of the proposed approach are effective in a comprehensive ablation study.

Acknowledgements: We thank Daniel Fried and Bryan Seybold for valuable discussions and feedback on the draft. This work was supported in part by DoD including DARPA's LwLL, PTG and/or SemaFor programs, as well as BAIR's industrial alliance programs.

References

1. Alayrac, J.B., Bojanowski, P., Agrawal, N., Sivic, J., Laptev, I., Lacoste-Julien, S.: Unsupervised learning from narrated instruction videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 4
2. Bojanowski, P., Lajugie, R., Bach, F., Laptev, I., Ponce, J., Schmid, C., Sivic, J.: Weakly supervised action labeling in videos under ordering constraints. In: European Conference on Computer Vision (ECCV) (2014) 4
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 9
4. Chang, C.Y., Huang, D.A., Sui, Y., Fei-Fei, L., Niebles, J.C.: D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4
5. De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., de Albuquerque Araújo, A.: Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. Patt. Rec. Letters (2011) 11
6. Ding, L., Xu, C.: Weakly-supervised action segmentation with iterative soft boundary assignment. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
7. Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D., Remagnino, P.: Summarizing videos with attention. Asian Conference on Computer Vision (ACCV) (2018) 4
8. Fried, D., Alayrac, J.B., Blunsom, P., Dyer, C., Clark, S., Nematzadeh, A.: Learning to segment actions from observation and narration. In: Association for Computational Linguistics (2020) 4
9. Gygli, M., Grabner, H., Riemenschneider, H., Gool, L.V.: Creating summaries from user videos. European Conference on Computer Vision (ECCV) (2014) 2, 4, 11
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 3
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 8
12. Huang, D.A., Fei-Fei, L., Niebles, J.C.: Connectionist temporal modeling for weakly supervised action labeling. In: European Conference on Computer Vision (ECCV) (2016) 4
13. Iashin, V., Rahtu, E.: A better use of audio-visual cues: Dense video captioning with bi-modal transformer. British Machine Vision Conference (BMVC) (2020) 11
14. Kanehira, A., Gool, L.V., Ushiku, Y., Harada, T.: Viewpoint-aware video summarization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
15. Kendall, M.G.: The treatment of ties in ranking problems. Biometrika **33**(3), 239–251 (1945) 3, 10
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR) (2015) 9
17. Kuehne, H., Richard, A., Gall, J.: Weakly supervised learning of actions from transcripts. In: CVIU (2017) 4
18. Kukleva, A., Kuehne, H., Sener, F., Gall, J.: Unsupervised learning of action classes with continuous temporal embedding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4

19. Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarization with adversarial lstm networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3
20. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 5, 6, 9, 10
21. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: IEEE International Conference on Computer Vision (ICCV) (2019) 2, 9
22. Narasimhan, M., Rohrbach, A., Darrell, T.: Clip-it! language-guided video summarization. Advances in Neural Information Processing Systems (NeurIPS) (2021) 2, 3, 4, 7, 9, 10, 11, 14
23. Open video project. https://open-video.org/ 11
24. Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J.: Rethinking the evaluation of video summaries. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 10
25. Park, J., Lee, J., Kim, I.J., Sohn, K.: Sumgraph: Video summarization via recursive graph modeling. European Conference on Computer Vision (ECCV) (2020) 3, 4, 7, 9
26. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021) 10
27. Richard, A., Kuehne, H., Gall, J.: Weakly supervised action learning with RNN based fine-to-coarse modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4
28. Richard, A., Kuehne, H., Gall, J.: Action sets: Weakly supervised action segmentation without ordering constraints. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
29. Rochan, M., Wang, Y.: Video summarization by learning from unpaired data. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 3, 10
30. Rochan, M., Ye, L., Wang, Y.: Video summarization using fully convolutional sequence networks. European Conference on Computer Vision (ECCV) (2018) 10
31. Sener, F., Yao, A.: Unsupervised learning and segmentation of complex activities from video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
32. Sener, O., Zamir, A.R., Savarese, S., Saxena, A.: Unsupervised semantic parsing of video collections. In: IEEE International Conference on Computer Vision (ICCV) (2015) 4
33. Sharghi, A., Gong, B., Shah, M.: Query-focused extractive video summarization. European Conference on Computer Vision (ECCV) (2016) 4
34. Sharghi, A., Laurel, J.S., Gong, B.: Query-focused video summarization: Dataset, evaluation, and a memory network based approach. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 2, 4
35. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: Tvsum: Summarizing web videos using titles. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 2, 4, 11
36. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 4, 7

37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 6000–6010 (2017) 7
38. Wei, H., Ni, B., Yan, Y., Yu, H., Yang, X., Yao, C.: Video summarization via semantic attended networks. The Association for the Advancement of Artificial Intelligence Conference (AAAI) (2018) 4
39. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: European Conference on Computer Vision (ECCV) (2018) 9
40. Yuan, L., Tay, F.E., Li, P., Zhou, L., Feng, J.: Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization. The Association for the Advancement of Artificial Intelligence Conference (AAAI) (2019) 3
41. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Summary transfer: Examplar-based subset selection for video summarization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 9
42. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Video summarization with long short-term memory. European Conference on Computer Vision (ECCV) (2016) 3
43. Zhang, K., Grauman, K., Sha, F.: Retrospective encoders for video summarization. European Conference on Computer Vision (ECCV) (2018) 3
44. Zhao, B., Li, X., Lu, X.: Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 3
45. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-task weakly supervised learning from instructional videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 4, 7
46. Zwillinger, D., Kokoska, S.: Crc standard probability and statistics tables and formulae. CRC Press (1999) 3, 10